CRM 89-127 / August 1989

AD-A221 090

RESEARCH MEMORANDUM

A USER'S GUIDE TO THE MOFO MODEL

James E. Schliessmann
Michael W. Price

A Division of CNA Hudson Institute

CENTER FOR NAVAL ANALYSES

4401 Ford Avenue • Post Office Box 16268 • Alexandria, Virginia 22302-0268

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION / AVAILABILITY OF REPORT Approved for Public Release; Distribution Unlimited
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) CRM 89-127	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Center for Naval Analyses 6b. OFFICE SYMBOL (If applicable) CNA	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State, and ZIP Code) 4401 Ford Avenue Alexandria, Virginia 22302-0268	7b. ADDRESS (City, State, and ZIP Code)
8a. NAME OF FUNDING ORGANIZATION Office of Naval Research 8b. OFFICE SYMBOL (If applicable) ONR	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER N00014-87-C-0001
8c. ADDRESS (City, State, and ZIP Code) 800 North Quincy Street Arlington, Virginia 22217	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO.: 65154N PROJECT NO.: R0148 TASK NO.: WORK UNIT ACCESSION NO.:

11. TITLE (Include Security Classification)
A User's Guide to the MOFO Model

12. PERSONAL AUTHOR(S)
James E. Schliessmann, Michael W. Price

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
Final	FROM TO	August 1989	95

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB-GROUP
05	02	
01	02	

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
Computer applications, Computer programs, Computerized simulation, Flight paths, Input output models, Instruction manuals, MOFO (model of flying objects), User manuals

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

This research memorandum is a user's guide for the six-degree-of-freedom fight path generator, MOFO (Model Of Flying Objects). Described in detail are the procedures for setting up, running, and modifying the model's underlying software, as well as input data requirements and the output that the model provides. A functional description of each program module is included, and listings of the FORTRAN source code and an automated modification routine appear as appendixes.

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT ☐ UNCLASSIFIED / UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE (Include Area Code) / 22c. OFFICE SYMBOL

DD FORM 1473, 84 MAR

83 APR edition may be used until exhausted.
All other editions are obsolete

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

CNA CENTER FOR NAVAL ANALYSES

A Division of Hudson Institute 4401 Ford Avenue • Post Office Box 16268 • Alexandria, Virginia 22302-0268 • (703) 824-2000

24 August 1989

MEMORANDUM FOR DISTRIBUTION LIST

Subj: Center for Naval Analyses Research Memorandum 89-127

Encl: (1) CNA Research Memorandum 89-127, *A User's Guide to the MOFO Model*, by James E. Schliessmann and Michael W. Price, Aug 1989

1. Enclosure (1) is forwarded as a matter of possible interest.

2. This Research Memorandum is a user's guide for the six-degree-of-freedom flight path generator, MOFO (Model Of Flying Objects). Described in detail are the procedures for setting up, running, and modifying the model's underlying software, as well as input data requirements and the output that the model provides. A functional description of each program module is included, and listings of the FORTRAN source code and an automated modification routine appear as appendixes.

Frank E. Schwamb
Director
Surface and Air Combat Systems Program

Distribution List:
Reverse page

Subj: Center for Naval Analyses Research Memorandum 89-127

Distribution List

SNDL

26F3	COMOPTEVFOR	
26KKK1	TACTRAGRULANT	
26KKK2	TACTRAGRUPAC	
FKQ6A	NAVAIRDEVCEN	
	Attn:	Mr. Ed. Lesoravage
FKQ6H	NAVWPNCEN	
	Attn:	Dr. P. Arnold

OPNAV

OP-05W/50W

CRM 89-127 / August 1989

A USER'S GUIDE TO THE MOFO MODEL

James E. Schliessmann
Michael W. Price

Advanced Systems Division

A Division of CNA Hudson Institute

CENTER FOR NAVAL ANALYSES

4401 Ford Avenue • Post Office Box 16268 • Alexandria, Virginia 22302-0268

ABSTRACT

This research memorandum is a user's guide for the six-degree-cf-freedom flight path generator, MOFO (Model Of Flying Objects). Described in detail are the procedures for setting up, running, and modifying the model's underlying software, as well as input data requirements and the output that the model provides. A functional description of each program module is included, and listings of the FORTRAN source code and an automated modification routine appear as appendixes.

CONTENTS

ILLUSTRATIONS

INTRODUCTION

Models used to simulate the inherent capabilities of modern defense systems are employed extensively by today's analyst. Because of rising costs on actual hardware testing and training, detailed models that provide reliable output are routinely sought after or developed. This is particularly true in the area of survivability analysis.

Combat survivability, by definition, is the ability of a vehicle to avoid and/or withstand a man-made hostile environment. The threat in such an environment consists of radars, guns, missiles, and so on. In survivability analysis, the effects of each threat in this hostile environment can be studied individually using sophisticated models. In aircraft combat survivability, a necessary input to these models is a position and attitude history of the aerodynamic vehicle being analyzed. The generation of such a flight path for the vehicle is therefore a necessary first step toward assessing the susceptibility and/or the vulnerability of the vehicle to a particular threat system.

MOFO (Model Of Flying Objects) is a computer model that generates flight paths of aerodynamic vehicles. It can simulate conventional aircraft and, with minor modifications, bank-to-turn missiles. The user provides both the necessary aerodynamic data and power parameters of the vehicle in one file and the commands that specify the intended flight path in another file. The model computes a six-degree-of-freedom trajectory consisting of a time history of position and attitude, using equations found in [1]. Such output can be used as input to models of radar detection or missile flyout as described in [2].

The purpose of this manual is to provide the user with the critical information necessary to set up and run the model. The first section provides a functional description of MOFO. It delineates the computational requirements and the model's coordinate systems. A second section covers setup and run instructions, and includes sample files and their respective formats. The manual concludes with improvements that should be implemented in the model at a later date.

FUNCTIONAL DESCRIPTION

COMPUTATIONAL DETAILS

MOFO is written in FORTRAN 77 and currently runs on a MicroVAX II computer. The present model requires roughly 350 blocks of memory on a Digital RA60 disk drive. This number includes all source code files, text and object libraries, the executable image, and ".COM" files. It does not include the storage requirement for either the input or output files, which is a function of the user's analytical objective and disk capacity. Except for a few long variable names that are allowed in the VAX/VMS FORTRAN language, standard coding conventions have been used. The code itself is unclassified, but the user may have classified input parameters to use. These attributes should serve to increase the transportability of MOFO to other machine environments.

In its present form, MOFO contains 1,300 lines of code divided into 23 separate modules: subroutines, functions, and the main program. Figure 1 provides an overview of the model's structure using a call tree diagram. Implicit double precision is used throughout in an effort to minimize round-off errors. Common blocks are placed in a separate file and an "INCLUDE" statement is used by the appropriate modules to incorporate the variable transfer structure. More specific information contained in appendix A describes both variable definitions and details of each module's function. MOFO is internally documented as well; the source code is contained in appendix B.

Whether the user is simply modifying a module of the model or creating the executable image from scratch, the CPU time required to compile and link the program is nominal. The user should structure the MOFO model directory as described in the file MODMOFO.COM (a hardcopy is located in appendix C). With this directory structure intact, the MODMOFO.COM file may be used to minimize the effort involved in the modification process. Actual run times for the model are, of course, dependent on the length of the scenario for which the model is trying to generate a trajectory. But the ratio of CPU time to real time is approximately one to one.

INERTIAL REFERENCE SYSTEM

MOFO models the flight of a single aerodynamic vehicle as it attempts to follow a commanded route via waypoints. Waypoints are positions in three-dimensional space that the aircraft attempts to fly through in an effort to produce the desired trajectory. The trajectory is generated in an inertial reference frame fixed on the surface of a flat earth. Position is described via Cartesian coordinates X, Y, and Z in this frame. These coordinates are measured in meters. As shown in figure 2, the positive X-axis is directed due east, the positive Y-axis due north, and the positive Z-axis upwards. The earth's atmosphere in MOFO is assumed stationary and homogeneous. The atmospheric pressure, density, and temperature profiles are those of the U.S. standard atmosphere. These profiles are computed from expressions found in [3].

LOCAL HORIZON

While MOFO is generating a flight path of positional data, it is also keeping track of the vehicle's attitude relative to the local horizon reference frame. The local horizon is a right-handed system with its origin based at the aircraft's center of gravity (c.g.). Each axis is parallel to its corresponding axis in the inertial reference frame (figure 2). To specify the aerodynamic forces and the attitude of the aircraft, two additional reference frames are needed: the wind system and the body system.

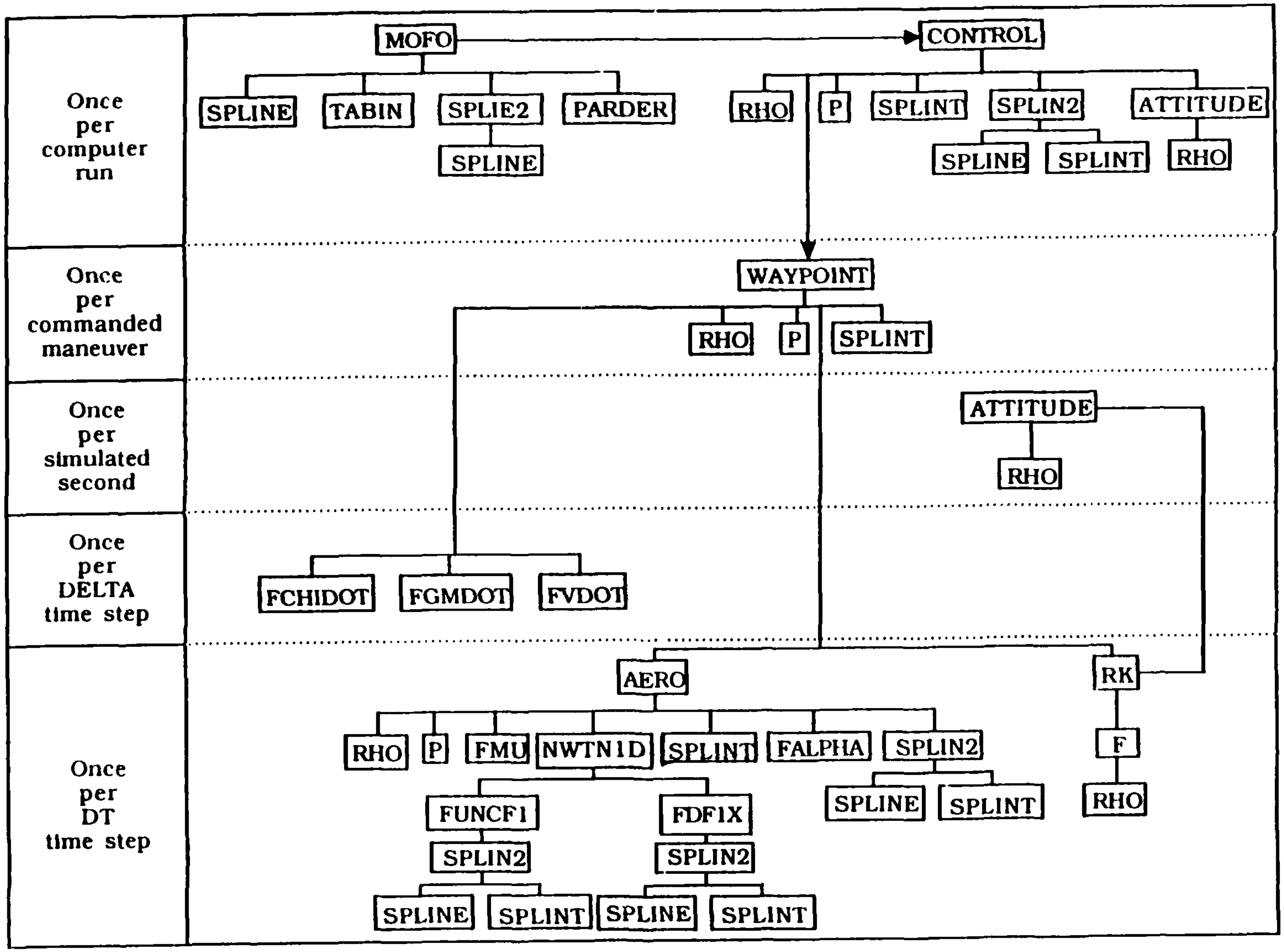

Figure 1. Call tree for MOFO modules

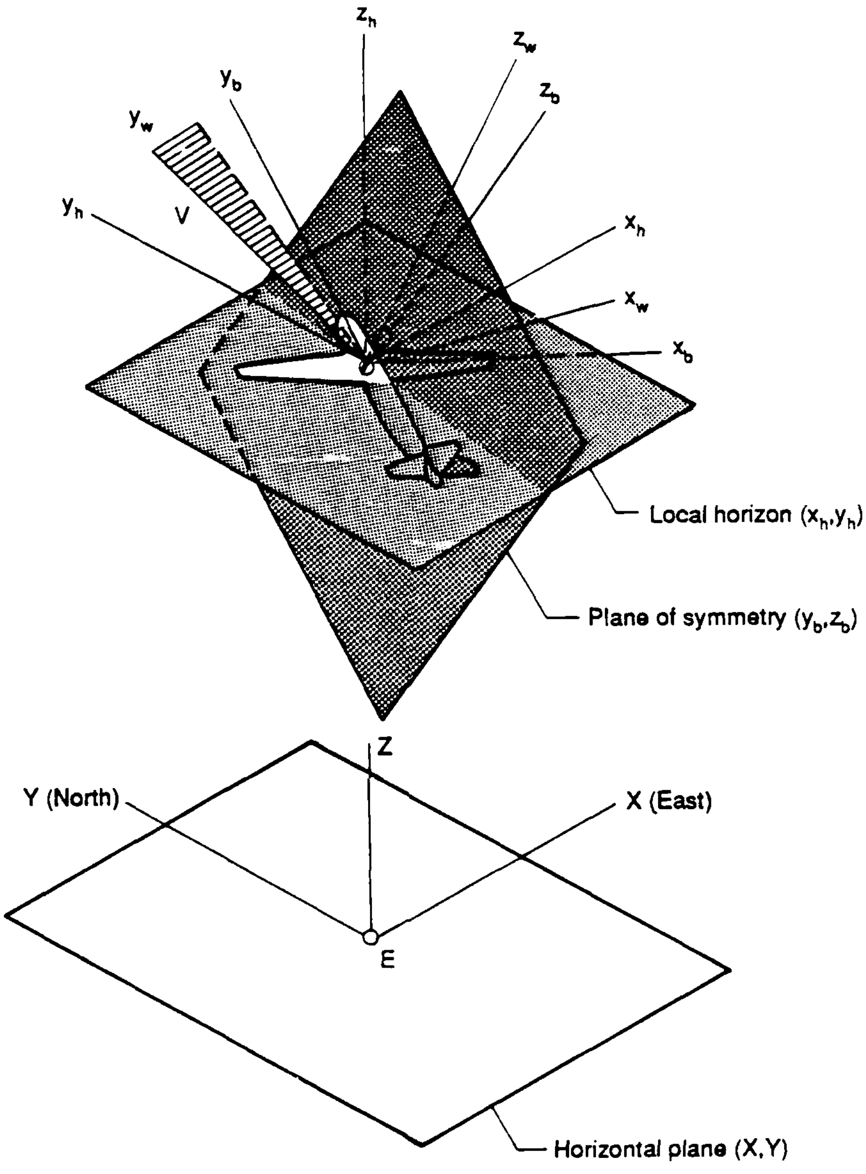

Figure 2. MOFO external coordinate system

WIND SYSTEM

The wind system specifies the direction of the aircraft's velocity vector and therefore serves as a natural frame of reference for describing the aerodynamic forces. Three rotations are needed to arrive at the wind system from the local horizon (see figure 3). Consider an aircraft that is flying straight and level with its velocity vector pointing northward along the y_h-axis and its starboard wing pointing eastward along the x_h-axis.

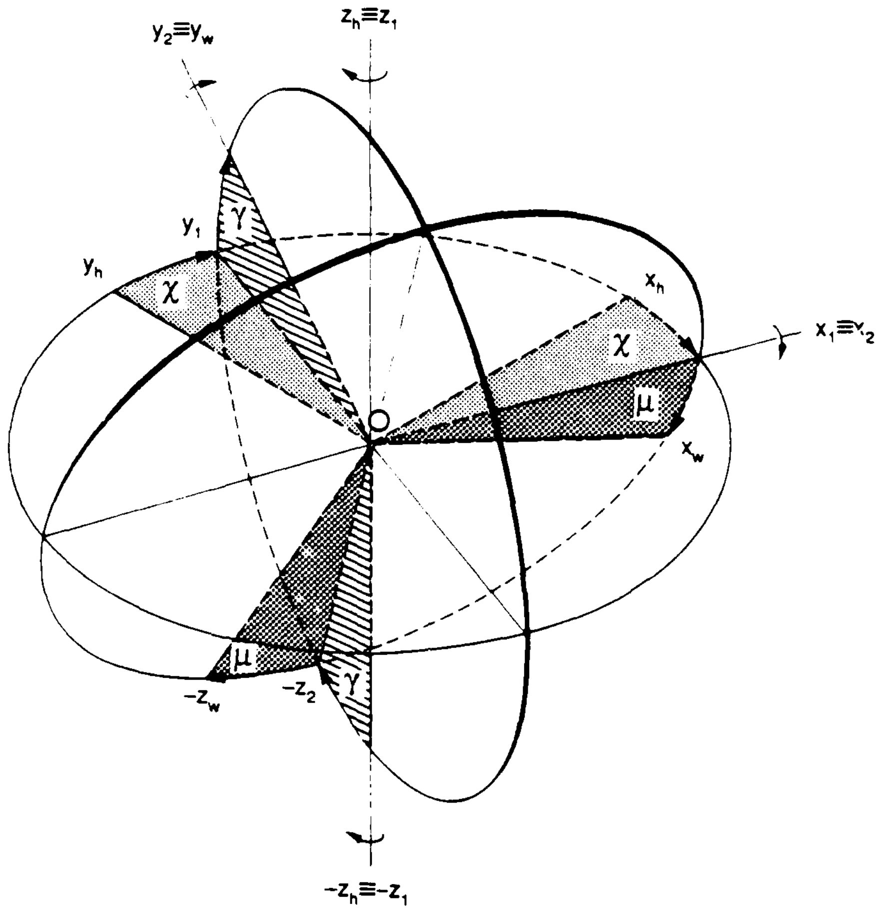

Figure 3. Rotations from the local horizon to the wind system

The following sequence of rotations about the c.g. is performed:

- The velocity vector is rotated in a left-handed sense through an angle χ about the z_h-axis in the x_h y_h plane to arrive at the intermediate x_1 y_1 z_1 system. The angle χ is the velocity heading relative to north.

- The velocity vector is then rotated in a right-handed sense through the angle γ about the x_1-axis in the y_1 z_1 plane to arrive at the intermediate x_2 y_2 z_2 system. The angle γ is the velocity pitch angle.

- Last, the aircraft is rotated in a right-handed sense through angle μ about the y_2-axis (velocity vector) in the x_2 z_2 plane to arrive at the x_w y_w z_w wind system. The angle μ is the velocity roll angle.

BODY SYSTEM

The body system is used to specify the orientation of the aircraft relative to the oncoming wind. Two angular rotations are needed to arrive at the body system from the wind system (see figure 4). The following sequence of rotations about the c.g. is performed:

- The aircraft is rotated in a left-handed sense through an angle σ about the z_w-axis in the x_w y_w plane to arrive at the intermediate x_3 y_3 z_3 system. The angle σ is the sideslip angle.

- The aircraft is then rotated in a right-handed sense through an angle α about the x_3-axis in the y_3 z_3 plane to arrive at the x_b y_b z_b body system. The angle α is the angle of attack.

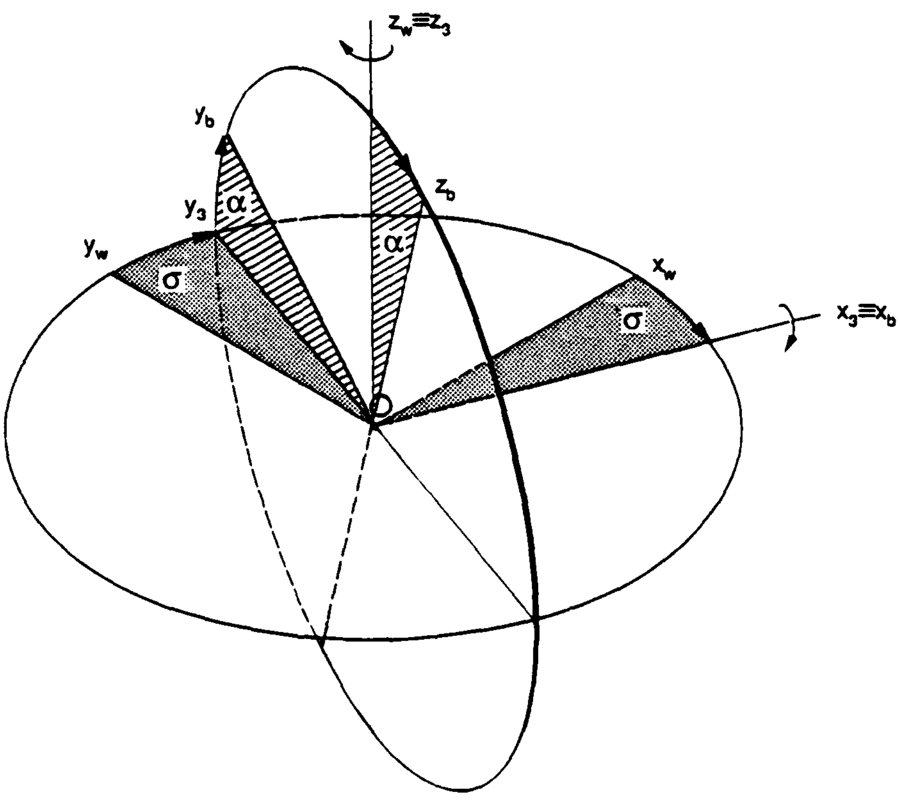

Figure 4. Rotations from the wind system to the body system

Because the aircraft is assumed to fly without sideslip, σ will always equal zero in MOFO. All the other angles (χ, γ, μ, and α) are written to an output file. These angles appear in the equations of motion as found in [1].

The body system can also be achieved directly from the local horizon in a manner analogous to the transformation to the wind system from the local horizon. The velocity angles in figure 3 (χ, γ, and μ) are replaced by the body angles (χ_b, γ_b, and μ_b, respectively). These angles are written to an output file along with the aircraft's positional data. This file is separate from the file that contains the velocity angles and the angle of attack. It is this file that would be used as input to other models requiring six-degree-of-freedom information. For example, a detailed radar model would need position and attitude information to determine the appropriate radar cross section (RCS) value along the radar's line of sight.

EXTERNAL VERSUS INTERNAL COORDINATE SYSTEMS

The external coordinate systems just described are used to ensure the compatibility of MOFO with other models used in the Surface and Air Combat Systems Program. The coordinate systems used internally in MOFO are analogous but slightly different (see [1]). For instance, the following axis inversions are required to transform the external inertial system to the internal inertial system:

$$Y_I = -Y_E \tag{1}$$

and

$$Z_I = -Z_E \tag{2}$$

where I denotes the internal system, and E denotes the external system as described under "Inertial Reference System." The subscript E was dropped before for simplicity of notation. These axis inversions create a right-handed coordinate system in the internal system. Because internal headings are measured from the inertial X_I-axis and external headings are from the inertial Y_E-axis, a third transformation is:

$$\chi_I = \chi_E - 90^\circ \quad . \tag{3}$$

Before the positional and angular data are written to the output files, the appropriate variables are transformed back to the external coordinate system for consistency. The user should be aware of these differences between the two systems in the event that later modifications are required.

SETUP AND RUN INSTRUCTIONS

DCL COMMAND RUN FILE

On the assumption that the user is familiar with running a ".COM" file interactively or in batch mode, the command file in the VAX/VMS operating system for running the MOFO model should resemble:

```
$ ASSIGN AIRCRAFT.IN FOR001
$ ASSIGN COMMAND.IN FOR002
$ !
$ ASSIGN POSITION.OUT FOR007
$ ASSIGN VELOCITY.OUT FOR008
$ ASSIGN ERROR.OUT FOR009
$ !
$ RUN MOFO.EXE
$ !
$ DEASSIGN FOR001
$ DEASSIGN FOR002
$ DEASSIGN FOR007
$ DEASSIGN FOR008
$ DEASSIGN FOR009
$ !
$ EXIT
```

Logical units 1 and 2 are the aircraft characteristics file and command file, respectively. Logical unit 9 contains error messages that alert the user to possible problems. Logical unit 7 is the output file that contains position and body attitude formation. Logical unit 8 is an output file that contains other information useful to the user, such as the velocity angles, angle of attack, and acceleration components. Each file is described in more detail below.

INPUT DATA

Input for MOFO consists of two files, an aircraft characteristics file and a command input file.

Aircraft Characteristics File

The aircraft characteristics file consists of both aerodynamic data and power data of the specific vehicle being modeled. The aerodynamic data consist of lift coefficients, drag coefficients, and aerodynamic time constants. The power data consist of thrust, fuel flow, and thrust-specific fuel consumption, among other things. Aircraft design variables are also required, such as the wing surface area, base weight of airframe, and maximum fuel capacity. Table 1 shows how this file is structured.

Users of the Advanced Air-to-Air System Performance Evaluation Model (AASPEM) may note the similarities in the input formats . Each line in table 1 corresponds to a record in the file, except for the representation of two-dimensional tables. All variables shown in the table are explained in detail in appendix A under the "Common Block" subsection.

Table 1. Aircraft input file format

Model variables	FORTRAN format
blank line for descriptive label	—
NMCHMX, NAOAMX	2I10
IALT(1)–IALT(NMCHMX)	10X,14I5
SPDMAX(1)–SPDMAX(NMCHMX)	10X,14F5.0
AIRSPD(1)–AIRSPD(NAOAMX)	10X,14F5.0
AOAMAX(1)–AOAMAX(NAOAMX)	10X,14F5.0
NALTML, NMCHML, SCALE	10X,2I5,30X,F10.0
blank line for table desciption	—
MCHMIL(1)–MCHMIL(NMCHML)	10X,14F5.0
ALTMIL(1),PWRMIL(1,1)–PWRMIL(1,NMCHML)	
.	.
.	.
.	.
ALTMIL(I),PWRMIL(I,1)–PWRMIL(I,NMCHML)	4X,I5,1X,14I5
.	.
.	.
.	.
ALTMIL(NALTML),PWRMIL(NALTML,1)–PWRMIL(NALTML,NMCHML)	
NALTAB,NMCHAB,SCALE	10X,2I5,30X,F10.0
blank line for table description	—
MCHAB(1)–MCHAB(NMCHAB)	10X,14F5.0
ALTAB(1),PWRAB(1,1)–PWRAB(1,NMCHAB)	
.	.
.	.
.	.
ALTAB(I),PWRAB(I,1)–PWRAB(I,NMCHAB)	4X,I5,1X,14I5
.	.
.	.
.	.
ALTAB(NALTAB),PWRAB(NALTAB,1)–PWRAB(NALTAB,NMCHAB)	
NALTML,NMCHML,SCALE	10X,2I5,30X,F10.0
blank line for table description	—
MCHMIL(1)–MCHMIL(NMCHML)	10X,14F5.0
ALTMIL(1),FULMIL(1,1)–FULMIL(1,NMCHML)	
.	.
.	.
.	.
ALTMIL(I),FULMIL(I,1)–FULMIL(I,NMCHML)	4X,I5,1X,14I5
.	.
.	.
.	.
ALTMIL(NALTML),FULMIL(NALTML,1)–FULMIL(NALTML,NMCHML)	
NALTAB,NMCHAB,SCALE	10X,2I5,30X,F10.0
blank line for table description	—
MCHAB(1)–MCHAB(NMCHAB)	10X,14F5.0

Table 1. (Continued)

Model variables	FORTRAN format
ALTAB(1),FULAB(1,1)–FULAB(1,NMCHAB)	
.	.
.	.
.	.
ALTAB(I),FULAB(I,1)–FULAB(I,NMCHAB)	4X,I5,1X,14I5
.	.
.	.
.	.
ALTAB(NALTAB),FULAB(NALTAB,1)–FULAB(NALTAB,NMCHAB)	
NCLCD,NMCHCD,SCALE	10X,2I5,30X,F10.0
blank line for table description	—
MACHCD(1)–MACHCD(NMCHCD)	10X,14F5.0
CLFTCD(1),CDRAG(1,1)–CDRAG(1,NMCHCD)	
.	.
.	.
.	.
CLFTCD(I),CDRAG(I,1)–CDRAG(I,NMCHCD)	4X,F5.0,1X,14F5.0
.	.
.	.
.	.
CLFTCD(NCLCD),CDRAG(NCLCD,1)–CDRAG(NCLCD,NMCHCD)	
NAOACL,NMCHCL,SCALE	10X,2I5,30X,F10.0
blank line for table description	—
MACHCL(1)–MACHCL(NMCHCL)	10X,14F5.0
AOACL(1),CLIFT(1,1)–CLIFT(1,NMCHCL)	
.	.
.	.
.	.
AOACL(I),CLIFT(I,1)–CLIFT(I,NMCHCL)	4X,F5.0,1X,14F5.0
.	.
.	.
.	.
AOACL(NAOACL),CLIFT(NAOACL,1)–CLIFT(NAOACL,NMCHCL)	
NCTRPM	I10
PCTRPM(1)–PCTRPM(NCTRPM)	10X,14F5.0
TSFC(1)–TSFC(NCTRPM)	10X,14F5.0
blank line for variable description	—
DT,DELTA,KAY	3F10.0
blank line for variable description	—
AOATAU,MUTAU,GMTAU,CHITAU,VTAU	5F10.0
blank line for variable description	—
AREA,WTAC,WTFLMX	3F15.0

MOFO was developed specifically for modeling military turbojet aircraft. The user may find that many of the input parameters are easily accessible for fighter and attack aircraft. NATOPS and TAC manuals are good initial starting points. These texts can be used to extract such basic information as the planform area (AREA), base weight (WTAC), and maximum internal fuel storage (WTFLMX). On the other hand, aero-dynamic coefficients and engine-specific thrust and fuel flow data are harder to come by. This information is usually obtained from the appropriate manufacturer or NAVAIR. Finally, there are variables that must be adjusted until output performance matches the user's expectation or validation data. Among this class of input variables are the integration time step, command guidance time step, and the proportional navigation constant (see [1] for details). Figure 5 provides the user with sample aircraft input file for the F/A-18 hornet.

Command Input File

The command input file consists of the aircraft's initial position followed by a series of instructions. These instructions constitute the commanded route on which the trajectory is generated. Such commands as what position to fly to or what heading to follow and with what climb or dive angle provide the user flexibility and control in prescribing a desired behavior. Users familiar with BLUEMAX II may notice similarities between this file and its flight control file used in the automated mode. The format for this input file is provided in table 2.

The file is composed of three parts. The first part describes the initial position, the direction and magnitude of the velocity vector, the angle of attack, and weight considerations. The first line provides the user with a line for a flight path description. A blank line then follows. Next, the initial Cartesian position of the aircraft and its speed are provided. The next record has the velocity angles and the angle of attack. The final record contains the initial time of the simulation and the fuel and armament weights.

The second part consists of a series of commands that maneuver the aircraft. Currently, MOFO has one instruction named WAYPT. WAYPT flies the aircraft to the next user-specified position along a path that is aerodynamically feasible. The user also specifies the speed to be attained when the waypoint is achieved and a g limit to impose constraints on directional changes. Future commands to be implemented are discussed in detail in the final section.

The third and final part of this input file is the END command. It concludes the flight path generation and stops program execution. Figure 6 shows a sample command input file.

OUTPUT DATA

Output from the model is contained in two files. One file (named POSITION.OUT in the sample ".COM" file) provides the user with simulation time, aircraft positions (X, Y, and Z), speed, and body attitude (χ_b, γ_b, and μ_b). Figure 7 is a sample of this output using the format (1X, 5F11.2, 3F11.4).

The second output file (named VELOCITY.OUT in the sample ".COM" file) provides the user with other useful information on the aircraft. This file contains weight, velocity angles (χ, γ, and μ), angle of attack (α), thrust, and the acceleration components (longitudinal, pitch, yaw, and total accelerations). Figure 8 is a sample of this output using the format (1X, 11F10.2).

```
F/A-18 AIRCRAFT CHARACTERISTIC FILE
12      12
           0 5000100001500020000250003000035000400004500050000 55000
         1.1 1.24 1.38 1.51 1.65 1.74 1.87 1.98 1.98 1.99  2.0  2.0
         0.2  0.6  0.8 0.85  0.9 0.95 1.05  1.1  1.2  1.4  1.6  1.8
        36.4 35.5 35.8 35.4 35.0 32.0 21.2 19.8 16.8 14.6 14.8 14.6
          12   10                                            2.0
THRUST, MILITARY POWER              POUNDS
          0.  0.2  0.4  0.6  0.8  1.0  1.2  1.4  1.6  1.8
    0  9556 9541 9693 9549 9370 7895 4259  523-1576-6745
 5000  8255 8233 8646 8723 8868 8585 5772 2761  496-3620
10000  7304 7322 7704 7788 8031 7974 6947 4439 2362  -77
15000  6111 6140 6497 6845 7159 7313 6648 5411 3089 1001
20000  5025 5057 5410 5729 6265 6514 6287 5888 4212 1719
25000  4035 4085 4415 4738 5297 5694 5740 5680 4781 2262
30000  3213 3230 3537 3849 4365 4907 5081 5372 5124 3327
35000  2559 2575 2795 3050 3548 4014 4363 4816 4834 3808
40000  2010 2001 2145 2351 2746 3131 3457 3847 3906 3195
45000  1625 1619 1663 1781 2096 2403 2679 3000 3045 2486
50000  1284 1288 1331 1404 1585 1826 2070 2306 2358 1912
55000   951  970 1031 1115 1253 1399 1562 1770 1816 1464
          12   10                                  2.0
THRUST, AFTER BURNER                POUNDS
          0.  0.2  0.4  0.6  0.8  1.0  1.2  1.4  1.6  1.8
    0 14398148981572116310171651643114503125371074 8-4118
 5000 12423128391398314760159281697015752148371322310087
10000 10943113251234813095142451541516584155611419013233
15000  9171 614010415114171258013861151321553313614 12762
20000  5750  208 5854 954410923121971367814958137941 1996
25000  6143 6410   79  -99 9166105771210113503132751 1174
30000  4954 5139 5818 4204  119 8961104721206212627 11278
35000  3991 4147 4630 5207 3856 7306 883010418111821 0695
40000  3138 3227  413 4022  385 3429 6978 8278 8945 8692
45000  2487 2585 2757  397 2286 4425 5425 6441 6966 6762
50000  1973 2047 2188 2391 2789 3376 4182 4963 5394 5245
55000  1438 1524 1677 1876 2176 2569 3165 3803 4086 4047
          12   10                                  2.0
FUEL FLOW, MILITARY POWER           POUNDS/HOUR
          0.  0.2  0.4  0.6  0.8  1.0  1.2  1.4  1.6  1.8
    0  7963 8523 9310 99291049210277 9945 9480 9123 8740
 5000  6861 7345 8189 8894 97311039710201 9705 9332 8978
10000  6081 6509 7243 7804 8595 934510105 9696 9015 9174
15000  5053 5414 6042 6779 7495 8313 9069 9347 8264 8300
20000  4145 4440 4995 5611 6462 7239 8085 8708 8082 7420
25000  3299 3559 4045 4597 5386 6218 7087 7750 7668 6633
30000  2593 2786 3196 3699 4380 5261 6099 6841 7107 6486
35000  2029 2182 2495 2891 3527 4234 5155 5902 6255 6105
40000  1599 1697 1924 2238 2739 3323 4098 4717 5029 4970
45000  1318 1396 1521 1722 2118 2583 3207 3698 3947 3900
50000  1063 1132 1241 1392 1635 2000 2498 2885 3094 3056
55000   801  869  980 1124 1326 1563 1925 2251 2421 2391
          12   10                                  2.0
FUEL FLOW, AFTER BURNER             POUNDS/HOUR
          0.  0.2  0.4  0.6  0.8  1.0  1.2  1.4  1.6  1.8
    0 28575306983377036798404264196644034440294369 4 8881
 5000 24251260082922732243363524059042943446374434644116
10000 21008225162527127777314123552940707426154325044678
15000 17445 541420940236732686830850353503918338801 40496
20000  8298  607 645919442227272626830637351263585335725
25000 1175312601  587  5741871022108261733017532286 31470
```

Figure 5. Sample aircraft characteristics input file

```
 30000  95801021511671 4925  5881835822029258722870728944
 35000  7806 8322 937410577  4682146771823021724243082573 5
 40000  6248 6584  635 8273  623 442714463173001945420772
 45000  5075 5395 5809  643 2969 905611353136031530716334
 50000  4128 4379 4726 5168 5894 7065 888910639120381283 1
 55000  3184 3413 3757 4178 4749 5549 6888 8340 899110072
          12   11                                    1.0
     DRAG COEFFICIENTS
          0.0 0.85  0.9 0.95  1.0 1.05  1.2  1.4  1.6  1.8  2.0
   0.0 .0276.0237.0264.0368.0510.0531.0539.0563.0571.0580.0587
  0.05 .0272.0234.0261.0367.0500.0519.0525.0543.0548.0551.0552
   0.1 .0276.0237.0264.0368.0502.0519.0526.0546.0552.0557.0558
   0.2 .0300.0261.0287.0388.0528.0552.0567.0603.0628.0657.0684
   0.3 .0348.0309.0334.0445.0613.0625.0667.0733.0796.0861.0928
   0.4 .0424.0386.0416.0557.0742.0751.0812.0908.1016.1136.1240
   0.5 .0526.0499.0534.0712.0922.0941.1012.1158.1321.1486.1666
   0.6 .0662.0651.0704.0927.1147.1171.1277.1493.1716.1946.2175
   0.8 .1028.1101.1274.1522.1742.1771.1992.2393.2801.2871.3245
   1.0 .1466.1984.2184.2407.2602.2646.3032.3763.2801.2871.3245
   1.2 .1202.3264.3454.3697.3932.4021.4497.4143.2801.2871.3245
  1.35 .3502.4524.4724.4987.5242.5356.4947.4143.2801.2871.3245
          11   12                                    1.0
     LIFT COEFFICIENTS
          0.2  0.6  0.8 0.85  0.9 0.95 1.05  1.1  1.2  1.4  1.6  1.8
   0.0  -.01 -.02 -.03 -.02 -.05 -.04 0.00 0.00 0.00 0.00  .01  .01
   4.0   .33  .39  .44  .47  .51  .51  .43  .34  .32  .28  .24  .22
   8.0   .69  .81  .85  .89  .92  .88  .74  .61  .61  .53  .44  .41
  12.0  1.04 1.15 1.08 1.05 1.08 1.18 1.00  .86  .87  .75  .63  .60
  16.0  1.22 1.29 1.18 1.17 1.19 1.35 1.21 1.10 1.10  .95  .80  .77
  20.0  1.36 1.42 1.29 1.28 1.30 1.46 1.35 1.31 1.30  .95  .80  .77
  24.0  1.49 1.54 1.43 1.39 1.40 1.53 1.48 1.31 1.30  .95  .80  .77
  28.0  1.61 1.66 1.55 1.52 1.53 1.57 1.48 1.31 1.30  .95  .80  .77
  32.0  1.76 1.78 1.63 1.61 1.60 1.58 1.48 1.31 1.30  .95  .80  .77
  36.0  1.82 1.81 1.69 1.64 1.61 1.57 1.48 1.31 1.30  .95  .80  .77
  40.0  1.80 1.76 1.62 1.60 1.59 1.57 1.48 1.31 1.30  .95  .80  .77
     8
        .325   .4   .5   .6   .7   .8   .9  1.0
        2.75  2.4 1.95  1.6  1.3  1.1  .99  1.0
DT(sec)   DELTA(sec)    KAY
  0.1000     1.0000   16.0000
AOATAU(sec) MUTAU(sec) GMTAU(sec) CHITAU(sec) VTAU(sec)
  .10000      .1000    1.0000     1.0000     1.0000
PLANFORM AREA(sqft) WTAC(lbs)    WTFLMX(lbs)
     400.0000        23500.00        10810.0
```

Figure 5. (Continued)

Table 2. Command file format

Model variables	Units	FORTRAN format
blank line for profile label	—	—
blank line	—	—
X0,Y0,Z0,V0	m,m,m,m/sec	4F10.0
CHI0,GM0,MU0,ALPHA0	deg,deg,deg,deg	4F10.0
T0,WTFUEL,BINGO,WTARM	sec,lb,lb,lb	4F10.0
.	.	.
.	.	.
.	.	.
blank line	—	—
WAYPT	command name only	A
XF,YF,ZF,VF,GEES	m,m,m,m/sec,—	5F10.0
.	.	.
.	.	.
.	.	.
blank line	—	—
END	command to end	A

```
          SAMPLE RUN FOR F/A-18

   -10000.0        0.0     172.24     139.01
       90.0        0.0        0.0        4.0
        0.0    20000.0        0.0        0.0

WAYPOINT
        0.0        0.0     172.24     139.01        5.0

WAYPOINT
        0.0    10000.0     172.24     139.01        2.0

WAYPOINT
    10000.0    10000.0     172.24     139.01        5.0

END
```

Figure 6. Sample command input file

t (s),	X (m),	Y (m),	Z (m),	V (m/s),	BODY YAW (deg.),	BODY PITCH (deg.),	BODY ROLL (deg.)
0.00	-10000.00	0.00	172.24	139.01	90.0000	4.0000	0.0000
1.00	-9860.99	0.00	172.26	139.01	90.0000	3.8948	0.0000
2.00	-9721.98	0.00	172.27	139.01	90.0000	3.8927	0.0000
3.00	-9582.97	0.00	172.29	139.01	90.0000	3.8911	0.0000
4.00	-9443.96	0.00	172.30	139.01	90.0000	3.8897	0.0000
5.00	-9304.95	0.00	172.30	139.01	90.0000	3.8887	0.0000
6.00	-9165.94	0.00	172.31	139.01	90.0000	3.8879	0.0000
7.00	-9026.93	0.00	172.31	139.01	90.0000	3.8872	0.0000
8.00	-8887.92	0.00	172.32	139.01	90.0000	3.8867	0.0000
9.00	-8748.91	0.00	172.32	139.01	90.0000	3.8864	0.0000
10.00	-8609.90	0.00	172.32	139.01	90.0000	3.8860	0.0000
11.00	-8470.89	0.00	172.32	139.01	90.0000	3.8858	0.0000
12.00	-8331.88	0.00	172.32	139.01	90.0000	3.8856	0.0000
13.00	-8192.87	0.00	172.32	139.01	90.0000	3.8854	0.0000
14.00	-8053.86	0.00	172.32	139.01	90.0000	3.8852	0.0000
15.00	-7914.85	0.00	172.31	139.01	90.0000	3.8850	0.0000
16.00	-7775.84	0.00	172.31	139.01	90.0000	3.8848	0.0000
17.00	-7636.83	0.00	172.31	139.01	90.0000	3.8847	0.0000
18.00	-7497.82	0.00	172.31	139.01	90.0000	3.8845	0.0000
19.00	-7358.81	0.00	172.31	139.01	90.0000	3.8844	0.0000
20.00	-7219.80	0.00	172.31	139.01	90.0000	3.8842	0.0000
21.00	-7080.79	0.00	172.31	139.01	90.0000	3.8841	0.0000
22.00	-6941.78	0.00	172.31	139.01	90.0000	3.8839	0.0000
23.00	-6802.77	0.00	172.31	139.01	90.0000	3.8837	0.0000
24.00	-6663.76	0.00	172.30	139.01	90.0000	3.8836	0.0000
25.00	-6524.75	0.00	172.30	139.01	90.0000	3.8834	0.0000
26.00	-6385.74	0.00	172.30	139.01	90.0000	3.8833	0.0000
27.00	-6246.73	0.00	172.30	139.01	90.0000	3.8831	0.0000
28.00	-6107.72	0.00	172.30	139.01	90.0000	3.8830	0.0000
29.00	-5968.71	0.00	172.30	139.01	90.0000	3.8828	0.0000
30.00	-5829.70	0.00	172.30	139.01	90.0000	3.8827	0.0000
31.00	-5690.69	0.00	172.30	139.01	90.0000	3.8825	0.0000
32.00	-5551.68	0.00	172.29	139.01	90.0000	3.8824	0.0000
33.00	-5412.67	0.00	172.29	139.01	90.0000	3.8822	0.0000
34.00	-5273.66	0.00	172.29	139.01	90.0000	3.8821	0.0000
35.00	-5134.65	0.00	172.29	139.01	90.0000	3.8819	0.0000
36.00	-4995.64	0.00	172.29	139.01	90.0000	3.8818	0.0000
37.00	-4856.63	0.00	172.29	139.01	90.0000	3.8816	0.0000
38.00	-4717.62	0.00	172.29	139.01	90.0000	3.8814	0.0000
39.00	-4578.61	0.00	172.29	139.01	90.0000	3.8813	0.0000
40.00	-4439.60	0.00	172.28	139.01	90.0000	3.8811	0.0000
41.00	-4300.59	0.00	172.28	139.01	90.0000	3.8810	0.0000
42.00	-4161.58	0.00	172.28	139.01	90.0000	3.8808	0.0000
43.00	-4022.57	0.00	172.28	139.01	90.0000	3.8807	0.0000
44.00	-3883.56	0.00	172.28	139.01	90.0000	3.8805	0.0000
45.00	-3744.55	0.00	172.28	139.01	90.0000	3.8804	0.0000
46.00	-3605.54	0.00	172.28	139.01	90.0000	3.8802	0.0000
47.00	-3466.53	0.00	172.28	139.01	90.0000	3.8801	0.0000
48.00	-3327.52	0.00	172.27	139.01	90.0000	3.8799	0.0000
49.00	-3188.51	0.00	172.27	139.01	90.0000	3.8798	0.0000
50.00	-3049.50	0.00	172.27	139.01	90.0000	3.8796	0.0000
51.00	-2910.49	0.00	172.27	139.01	90.0000	3.8794	0.0000

Figure 7. Sample position and attitude output

52.00	-2771.48	0.00	172.27	139.01	90.0000	3.8793	0.0000
53.00	-2632.47	0.00	172.27	139.01	90.0000	3.8791	0.0000
54.00	-2493.46	0.00	172.27	139.01	90.0000	3.8790	0.0000
55.00	-2354.45	0.00	172.26	139.01	90.0000	3.8788	0.0000
56.00	-2215.44	0.00	172.26	139.01	90.0000	3.8787	0.0000
57.00	-2076.43	0.00	172.26	139.01	90.0000	3.8785	0.0000
58.00	-1937.42	0.00	172.26	139.01	90.0000	3.8784	0.0000
59.00	-1798.41	0.00	172.26	139.01	90.0000	3.8782	0.0000
60.00	-1659.40	0.00	172.26	139.01	90.0000	3.8781	0.0000
61.00	-1520.39	0.00	172.26	139.01	90.0000	3.8779	0.0000
62.00	-1381.38	0.00	172.25	139.01	90.0000	3.8778	0.0000
63.00	-1242.37	0.00	172.25	139.01	90.0000	3.8776	0.0000
64.00	-1103.36	0.00	172.25	139.01	90.0000	3.8775	0.0000
65.00	-964.35	0.00	172.25	139.01	90.0000	3.8773	0.0000
66.00	-825.34	0.00	172.25	139.01	90.0000	3.8771	0.0000
67.00	-686.33	0.00	172.25	139.01	90.0000	3.8770	0.0000
68.00	-547.32	0.00	172.25	139.01	90.0000	3.8768	0.0000
69.00	-408.31	0.00	172.24	139.01	90.0000	3.8767	0.0000
70.00	-269.30	0.00	172.24	139.01	90.0000	3.8765	0.0000
71.00	-130.29	0.00	172.24	139.01	90.0000	3.8764	0.0000
72.00	8.68	2.31	172.12	139.01	82.2859	3.5927	-44.9966
73.00	146.87	16.70	171.82	139.01	74.4440	3.4834	-49.6857
74.00	282.29	47.69	171.45	139.01	66.4462	3.4564	-50.7880
75.00	412.34	96.47	171.07	139.01	58.3946	3.4528	-51.1322
76.00	534.42	162.72	170.70	139.01	50.3236	3.4596	-51.2471
77.00	646.05	245.37	170.34	139.01	42.2452	3.4721	-51.2836
78.00	744.98	342.86	170.02	139.01	34.1640	3.4888	-51.2924
79.00	829.24	453.29	169.74	139.01	26.0819	3.5089	-51.2915
80.00	897.20	574.42	169.50	139.01	18.2197	3.5378	-51.0907
81.00	948.18	703.65	169.35	139.01	11.6588	3.6022	-49.6423
82.00	983.41	838.05	169.31	139.01	6.4352	3.6859	-47.0076
83.00	1005.26	975.28	169.38	139.01	2.4376	3.7773	-42.9941
84.00	1016.38	1113.82	169.54	139.01	359.4944	3.8524	-37.5449
85.00	1019.30	1252.78	169.75	139.01	357.3916	3.9019	-31.0223
86.00	1016.13	1391.75	169.99	139.01	355.9238	3.9260	-24.2096
87.00	1008.58	1530.55	170.22	139.01	354.9191	3.9316	-17.9473
88.00	997.94	1669.15	170.42	139.01	354.2438	3.9267	-12.7556
89.00	985.16	1807.57	170.59	139.01	353.7989	3.9176	-8.7623
90.00	970.94	1945.85	170.73	139.01	353.5125	3.9081	-5.8468
91.00	955.76	2084.03	170.84	139.01	353.3328	3.8998	-3.7963
92.00	939.97	2222.14	170.92	139.01	353.2238	3.8934	-2.3960
93.00	923.79	2360.20	170.98	139.01	353.1603	3.8887	-1.4648
94.00	907.37	2498.24	171.02	139.01	353.1252	3.8855	-0.8618
95.00	890.81	2636.26	171.06	139.01	353.1073	3.8834	-0.4826
96.00	874.17	2774.27	171.09	139.01	353.0993	3.8821	-0.2524
97.00	857.49	2912.27	171.12	139.01	353.0967	3.8813	-0.1184
98.00	840.80	3050.28	171.14	139.01	353.0966	3.8808	-0.0449
99.00	824.09	3188.28	171.16	139.01	353.0977	3.8805	-0.0079
100.00	807.39	3326.28	171.19	139.01	353.0989	3.8803	0.0082
101.00	790.69	3464.29	171.21	139.01	353.1000	3.8802	0.0130
102.00	773.98	3602.29	171.23	139.01	353.1009	3.8801	0.0125
103.00	757.29	3740.29	171.25	139.01	353.1015	3.8800	0.0098
104.00	740.59	3878.30	171.27	139.01	353.1018	3.8798	0.0068
105.00	723.89	4016.30	171.30	139.01	353.1020	3.8797	0.0042
106.00	707.20	4154.30	171.32	139.01	353.1021	3.8796	0.0023
107.00	690.50	4292.31	171.34	139.01	353.1021	3.8794	0.0011

Figure 7. (Continued)

108.00	673.80	4430.31	171.36	139.01	353.1021	3.8793	0.0003
109.00	657.11	4568.31	171.38	139.01	353.1021	3.8792	0.0000
110.00	640.41	4706.32	171.40	139.01	353.1021	3.8790	-0.0002
111.00	623.72	4844.32	171.43	139.01	353.1021	3.8789	-0.0002
112.00	607.02	4982.32	171.45	139.01	353.1020	3.8787	-0.0002
113.00	590.33	5120.33	171.47	139.01	353.1020	3.8786	-0.0001
114.00	573.63	5258.33	171.49	139.01	353.1020	3.8784	-0.0001
115.00	556.94	5396.34	171.51	139.01	353.1020	3.8783	0.0000
116.00	540.24	5534.34	171.54	139.01	353.1020	3.8781	0.0000
117.00	523.55	5672.34	171.56	139.01	353.1020	3.8780	0.0000
118.00	506.85	5810.35	171.58	139.01	353.1020	3.8778	0.0000
119.00	490.16	5948.35	171.60	139.01	353.1020	3.8777	0.0000
120.00	473.46	6086.35	171.62	139.01	353.1020	3.8776	0.0000
121.00	456.77	6224.36	171.64	139.01	353.1020	3.8774	0.0000
122.00	440.07	6362.36	171.67	139.01	353.1020	3.8773	0.0000
123.00	423.38	6500.37	171.69	139.01	353.1020	3.8771	0.0000
124.00	406.68	6638.37	171.71	139.01	353.1020	3.8770	0.0000
125.00	389.99	6776.37	171.73	139.01	353.1020	3.8768	0.0000
126.00	373.29	6914.38	171.75	139.01	353.1020	3.8767	0.0000
127.00	356.59	7052.38	171.78	139.01	353.1020	3.8765	0.0000
128.00	339.90	7190.38	171.80	139.01	353.1020	3.8764	0.0000
129.00	323.20	7328.39	171.82	139.01	353.1020	3.8762	0.0000
130.00	306.51	7466.39	171.84	139.01	353.1020	3.8761	0.0000
131.00	289.81	7604.40	171.86	139.01	353.1020	3.8759	0.0000
132.00	273.12	7742.40	171.88	139.01	353.1020	3.8758	0.0000
133.00	256.42	7880.40	171.91	139.01	353.1020	3.8757	0.0000
134.00	239.73	8018.41	171.93	139.01	353.1020	3.8755	0.0000
135.00	223.03	8156.41	171.95	139.01	353.1020	3.8754	0.0000
136.00	206.34	8294.41	171.97	139.01	353.1020	3.8752	0.0000
137.00	189.64	8432.42	171.99	139.01	353.1020	3.8751	0.0000
138.00	172.95	8570.42	172.02	139.01	353.1020	3.8749	0.0000
139.00	156.25	8708.43	172.04	139.01	353.1020	3.8748	0.0000
140.00	139.56	8846.43	172.06	139.01	353.1020	3.8746	0.0000
141.00	122.86	8984.43	172.08	139.01	353.1020	3.8745	0.0000
142.00	106.17	9122.44	172.10	139.01	353.1020	3.8743	0.0000
143.00	89.47	9260.44	172.12	139.01	353.1020	3.8742	0.0000
144.00	72.77	9398.45	172.15	139.01	353.1020	3.8741	0.0000
145.00	56.08	9536.45	172.17	139.01	353.1020	3.8739	0.0000
146.00	39.38	9674.45	172.19	139.01	353.1020	3.8738	0.0000
147.00	22.69	9812.46	172.21	139.01	353.1020	3.8736	0.0000
148.00	6.44	9950.51	172.19	139.01	0.3508	3.2161	49.7423
149.00	2.83	10089.35	171.82	139.01	12.3136	3.2649	53.7562
150.00	23.36	10226.62	171.32	139.01	24.7295	3.3352	54.3592
151.00	71.20	10356.87	170.79	139.01	36.8428	3.3778	54.3684
152.00	145.02	10474.36	170.29	139.01	48.1031	3.4048	54.0381
153.00	240.82	10574.79	169.88	139.01	58.2052	3.4497	53.3502
154.00	353.39	10656.05	169.58	139.01	66.9603	3.5367	52.1758
155.00	477.56	10718.25	169.41	139.01	74.2025	3.6349	50.2904
156.00	609.03	10763.14	169.38	139.01	79.9633	3.7267	47.3960
157.00	744.66	10793.36	169.48	139.01	84.4165	3.8199	43.2120
158.00	882.40	10811.81	169.67	139.01	87.7435	3.8913	37.6750
159.00	1021.08	10821.15	169.91	139.01	90.1655	3.9330	31.1541
160.00	1160.05	10823.69	170.18	139.01	91.8935	3.9471	24.4077
161.00	1299.03	10821.25	170.44	139.01	93.1068	3.9434	18.2249
162.00	1437.91	10815.27	170.67	139.01	93.9454	3.9311	13.0860
163.00	1576.66	10806.79	170.86	139.01	94.5156	3.9166	9.1071

Figure 7. (Continued)

164.00	1715.29	10796.59	171.02	139.01	94.8961	3.9032	6.1744
165.00	1853.84	10785.23	171.14	139.01	95.1449	3.8920	4.0868
166.00	1992.32	10773.10	171.23	139.01	95.3037	3.8835	2.6403
167.00	2130.75	10760.46	171.29	139.01	95.4023	3.8772	1.6614
168.00	2269.16	10747.51	171.34	139.01	95.4616	3.8729	1.0140
169.00	2407.54	10734.36	171.38	139.01	95.4957	3.8699	0.5962
170.00	2545.92	10721.10	171.41	139.01	95.5143	3.8680	0.3341
171.00	2684.29	10707.77	171.43	139.01	95.5236	3.8668	0.1750
172.00	2822.65	10694.40	171.45	139.01	95.5276	3.8660	0.0825
173.00	2961.02	10681.02	171.47	139.01	95.5289	3.8655	0.0317
174.00	3099.38	10667.63	171.48	139.01	95.5288	3.8652	0.0060
175.00	3237.75	10654.24	171.50	139.01	95.5282	3.8650	-0.0052
176.00	3376.11	10640.84	171.51	139.01	95.5275	3.8649	-0.0087
177.00	3514.47	10627.45	171.53	139.01	95.5269	3.8648	-0.0084
178.00	3652.84	10614.06	171.54	139.01	95.5265	3.8646	-0.0067
179.00	3791.20	10600.68	171.56	139.01	95.5262	3.8645	-0.0047
180.00	3929.56	10587.29	171.57	139.01	95.5260	3.8644	-0.0029
181.00	4067.93	10573.90	171.59	139.01	95.5259	3.8642	-0.0016
182.00	4206.29	10560.52	171.60	139.01	95.5259	3.8641	-0.0008
183.00	4344.66	10547.13	171.62	139.01	95.5259	3.8639	-0.0002
184.00	4483.02	10533.74	171.63	139.01	95.5259	3.8638	0.0000
185.00	4621.38	10520.36	171.65	139.01	95.5259	3.8636	0.0001
186.00	4759.75	10506.97	171.67	139.01	95.5259	3.8635	0.0001
187.00	4898.11	10493.59	171.68	139.01	95.5259	3.8633	0.0001
188.00	5036.48	10480.20	171.70	139.01	95.5259	3.8632	0.0001
189.00	5174.84	10466.81	171.71	139.01	95.5259	3.8631	0.0000
190.00	5313.20	10453.43	171.73	139.01	95.5259	3.8629	0.0000
191.00	5451.57	10440.04	171.74	139.01	95.5259	3.8628	0.0000
192.00	5589.93	10426.66	171.76	139.01	95.5259	3.8626	0.0000
193.00	5728.30	10413.27	171.77	139.01	95.5259	3.8625	0.0000
194.00	5866.66	10399.88	171.79	139.01	95.5259	3.8623	0.0000
195.00	6005.02	10386.50	171.80	139.01	95.5259	3.8622	0.0000
196.00	6143.39	10373.11	171.82	139.01	95.5259	3.8620	0.0000
197.00	6281.75	10359.72	171.83	139.01	95.5259	3.8619	0.0000
198.00	6420.12	10346.34	171.85	139.01	95.5259	3.8617	0.0000
199.00	6558.48	10332.95	171.86	139.01	95.5259	3.8616	0.0000
200.00	6696.84	10319.57	171.88	139.01	95.5259	3.8614	0.0000
201.00	6835.21	10306.18	171.89	139.01	95.5259	3.8613	0.0000
202.00	6973.57	10292.79	171.91	139.01	95.5259	3.8611	0.0000
203.00	7111.94	10279.41	171.92	139.01	95.5259	3.8610	0.0000
204.00	7250.30	10266.02	171.94	139.01	95.5259	3.8608	0.0000
205.00	7388.66	10252.64	171.95	139.01	95.5259	3.8607	0.0000
206.00	7527.03	10239.25	171.97	139.01	95.5259	3.8605	0.0000
207.00	7665.39	10225.86	171.98	139.01	95.5259	3.8604	0.0000
208.00	7803.75	10212.48	172.00	139.01	95.5259	3.8602	0.0000
209.00	7942.12	10199.09	172.02	139.01	95.5259	3.8601	0.0000
210.00	8080.48	10185.71	172.03	139.01	95.5259	3.8600	0.0000
211.00	8218.85	10172.32	172.05	139.01	95.5259	3.8598	0.0000
212.00	8357.21	10158.93	172.06	139.01	95.5259	3.8597	0.0000
213.00	8495.57	10145.55	172.08	139.01	95.5259	3.8595	0.0000
214.00	8633.94	10132.16	172.09	139.01	95.5259	3.8594	0.0000
215.00	8772.30	10118.77	172.11	139.01	95.5259	3.8592	0.0000
216.00	8910.67	10105.39	172.12	139.01	95.5259	3.8591	0.0000
217.00	9049.03	10092.00	172.14	139.01	95.5259	3.8589	0.0000
218.00	9187.39	10078.62	172.15	139.01	95.5259	3.8588	0.0000
219.00	9325.76	10065.23	172.17	139.01	95.5259	3.8586	0.0000
220.00	9464.12	10051.84	172.18	139.01	95.5259	3.8585	0.0000
221.00	9602.49	10038.46	172.20	139.01	95.5259	3.8583	0.0000
222.00	9740.85	10025.07	172.21	139.01	95.5259	3.8582	0.0000
223.00	9879.21	10011.69	172.23	139.01	95.5259	3.8580	0.0000

Figure 7. (Continued)

t (s),	m (slugs),	GAMMA (deg.),	CHI (deg.),	AOA (deg.),	MU (deg.),	T (lbs.),	Al (m/s**2),	Ap (m/s**2),	Ay (m/s**2),	At (m/s**2)
0.00	1066.52	0.00	90.00	4.00	0.00	3343.06	0.00	0.03	0.00	0.03
1.00	1066.48	0.01	90.00	3.89	0.00	3256.35	0.00	0.00	0.00	0.00
2.00	1066.43	0.01	90.00	3.89	0.00	3254.98	0.00	0.00	0.00	0.00
3.00	1066.39	0.00	90.00	3.89	0.00	3253.92	0.00	0.00	0.00	0.00
4.00	1066.34	0.00	90.00	3.89	0.00	3253.10	0.00	0.00	0.00	0.00
5.00	1066.30	0.00	90.00	3.89	0.00	3252.46	0.00	0.00	0.00	0.00
6.00	1066.25	0.00	90.00	3.89	0.00	3251.97	0.00	0.00	0.00	0.00
7.00	1066.21	0.00	90.00	3.89	0.00	3251.58	0.00	0.00	0.00	0.00
8.00	1066.16	0.00	90.00	3.89	0.00	3251.28	0.00	0.00	0.00	0.00
9.00	1066.12	0.00	90.00	3.89	0.00	3251.03	0.00	0.00	0.00	0.00
10.00	1066.07	0.00	90.00	3.89	0.00	3250.82	0.00	0.00	0.00	0.00
11.00	1066.03	0.00	90.00	3.89	0.00	3250.64	0.00	0.00	0.00	0.00
12.00	1065.99	0.00	90.00	3.89	0.00	3250.48	0.00	0.00	0.00	0.00
13.00	1065.94	0.00	90.00	3.89	0.00	3250.34	0.00	0.00	0.00	0.00
14.00	1065.90	0.00	90.00	3.89	0.00	3250.20	0.00	0.00	0.00	0.00
15.00	1065.85	0.00	90.00	3.89	0.00	3250.08	0.00	0.00	0.00	0.00
16.00	1065.81	0.00	90.00	3.89	0.00	3249.95	0.00	0.00	0.00	0.00
17.00	1065.76	0.00	90.00	3.89	0.00	3249.83	0.00	0.00	0.00	0.00
18.00	1065.72	0.00	90.00	3.88	0.00	3249.71	0.00	0.00	0.00	0.00
19.00	1065.67	0.00	90.00	3.88	0.00	3249.59	0.00	0.00	0.00	0.00
20.00	1065.63	0.00	90.00	3.88	0.00	3249.47	0.00	0.00	0.00	0.00
21.00	1065.58	0.00	90.00	3.88	0.00	3249.35	0.00	0.00	0.00	0.00
22.00	1065.54	0.00	90.00	3.88	0.00	3249.22	0.00	0.00	0.00	0.00
23.00	1065.49	0.00	90.00	3.88	0.00	3249.10	0.00	0.00	0.00	0.00
24.00	1065.45	0.00	90.00	3.88	0.00	3248.98	0.00	0.00	0.00	0.00
25.00	1065.40	0.00	90.00	3.88	0.00	3248.86	0.00	0.00	0.00	0.00
26.00	1065.36	0.00	90.00	3.88	0.00	3248.74	0.00	0.00	0.00	0.00
27.00	1065.32	0.00	90.00	3.88	0.00	3248.62	0.00	0.00	0.00	0.00
28.00	1065.27	0.00	90.00	3.88	0.00	3248.50	0.00	0.00	0.00	0.00
29.00	1065.23	0.00	90.00	3.88	0.00	3248.38	0.00	0.00	0.00	0.00
30.00	1065.18	0.00	90.00	3.88	0.00	3248.26	0.00	0.00	0.00	0.00
31.00	1065.14	0.00	90.00	3.88	0.00	3248.14	0.00	0.00	0.00	0.00
32.00	1065.09	0.00	90.00	3.88	0.00	3248.02	0.00	0.00	0.00	0.00
33.00	1065.05	0.00	90.00	3.88	0.00	3247.90	0.00	0.00	0.00	0.00
34.00	1065.00	0.00	90.00	3.88	0.00	3247.78	0.00	0.00	0.00	0.00
35.00	1064.96	0.00	90.00	3.88	0.00	3247.66	0.00	0.00	0.00	0.00
36.00	1064.91	0.00	90.00	3.88	0.00	3247.54	0.00	0.00	0.00	0.00
37.00	1064.87	0.00	90.00	3.88	0.00	3247.42	0.00	0.00	0.00	0.00
38.00	1064.82	0.00	90.00	3.88	0.00	3247.30	0.00	0.00	0.00	0.00
39.00	1064.78	0.00	90.00	3.88	0.00	3247.18	0.00	0.00	0.00	0.00
40.00	1064.73	0.00	90.00	3.88	0.00	3247.06	0.00	0.00	0.00	0.00
41.00	1064.69	0.00	90.00	3.88	0.00	3246.94	0.00	0.00	0.00	0.00
42.00	1064.65	0.00	90.00	3.88	0.00	3246.82	0.00	0.00	0.00	0.00
43.00	1064.60	0.00	90.00	3.88	0.00	3246.70	0.00	0.00	0.00	0.00
44.00	1064.56	0.00	90.00	3.88	0.00	3246.58	0.00	0.00	0.00	0.00
45.00	1064.51	0.00	90.00	3.88	0.00	3246.46	0.00	0.00	0.00	0.00
46.00	1064.47	0.00	90.00	3.88	0.00	3246.34	0.00	0.00	0.00	0.00
47.00	1064.42	0.00	90.00	3.88	0.00	3246.22	0.00	0.00	0.00	0.00
48.00	1064.38	0.00	90.00	3.88	0.00	3246.10	0.00	0.00	0.00	0.00
49.00	1064.33	0.00	90.00	3.88	0.00	3245.98	0.00	0.00	0.00	0.00
50.00	1064.29	0.00	90.00	3.88	0.00	3245.86	0.00	0.00	0.00	0.00
51.00	1064.24	0.00	90.00	3.88	0.00	3245.74	0.00	0.00	0.00	0.00

Figure 8. Sample velocity output

52.00	1064.20	0.00	90.00	3.88	0.00	3245.62	0.00	0.00	0.00	0.00
53.00	1064.15	0.00	90.00	3.88	0.00	3245.50	0.00	0.00	0.00	0.00
54.00	1064.11	0.00	90.00	3.88	0.00	3245.38	0.00	0.00	0.00	0.00
55.00	1064.07	0.00	90.00	3.88	0.00	3245.26	0.00	0.00	0.00	0.00
56.00	1064.02	0.00	90.00	3.88	0.00	3245.14	0.00	0.00	0.00	0.00
57.00	1063.98	0.00	90.00	3.88	0.00	3245.02	0.00	0.00	0.00	0.00
58.00	1063.93	0.00	90.00	3.88	0.00	3244.90	0.00	0.00	0.00	0.00
59.00	1063.89	0.00	90.00	3.88	0.00	3244.78	0.00	0.00	0.00	0.00
60.00	1063.84	0.00	90.00	3.88	0.00	3244.66	0.00	0.00	0.00	0.00
61.00	1063.80	0.00	90.00	3.88	0.00	3244.54	0.00	0.00	0.00	0.00
62.00	1063.75	0.00	90.00	3.88	0.00	3244.42	0.00	0.00	0.00	0.00
63.00	1063.71	0.00	90.00	3.88	0.00	3244.30	0.00	0.00	0.00	0.00
64.00	1063.66	0.00	90.00	3.88	0.00	3244.18	0.00	0.00	0.00	0.00
65.00	1063.62	0.00	90.00	3.88	0.00	3244.07	0.00	0.00	0.00	0.00
66.00	1063.57	0.00	90.00	3.88	0.00	3243.95	0.00	0.00	0.00	0.00
67.00	1063.53	0.00	90.00	3.88	0.00	3243.83	0.00	0.00	0.00	0.00
68.00	1063.49	0.00	90.00	3.88	0.00	3243.71	0.00	0.00	0.00	0.00
69.00	1063.44	0.00	90.00	3.88	0.00	3243.59	0.00	0.00	0.00	0.00
70.00	1063.40	0.00	90.00	3.88	0.00	3243.47	0.00	0.00	0.00	0.00
71.00	1063.35	0.00	90.00	3.88	0.00	3243.35	0.00	0.00	0.00	0.00
72.00	1063.30	-0.10	86.96	5.95	-51.61	5661.36	0.00	0.00	-1.26	1.26
73.00	1063.23	-0.14	80.73	7.26	-59.95	5782.98	0.00	0.00	-1.73	1.73
74.00	1063.17	-0.16	73.33	7.77	-62.23	5607.13	0.00	0.00	-1.90	1.90
75.00	1063.11	-0.16	65.50	7.96	-62.97	5648.92	0.00	0.00	-1.96	1.96
76.00	1063.05	-0.15	57.50	8.03	-63.23	5689.84	0.00	0.00	-1.99	1.99
77.00	1062.98	-0.14	49.45	8.06	-63.31	5713.50	0.00	0.00	-2.00	2.00
78.00	1062.92	-0.13	41.38	8.07	-63.32	5729.22	0.00	0.00	-2.00	2.00
79.00	1062.86	-0.11	33.30	8.07	-63.32	5742.76	0.00	0.00	-2.00	2.00
80.00	1062.79	-0.08	25.31	7.96	-62.87	5692.77	0.00	0.01	-1.96	1.96
81.00	1062.73	-0.04	17.95	7.27	-59.85	5836.50	0.00	0.01	-1.73	1.73
82.00	1062.66	0.00	11.70	6.42	-54.96	6121.86	0.00	0.00	-1.43	1.43
83.00	1062.60	0.05	6.66	5.63	-48.48	5298.86	0.00	0.00	-1.13	1.13
84.00	1062.54	0.08	2.76	4.99	-40.83	4411.61	0.00	0.00	-0.86	0.86
85.00	1062.49	0.09	-0.16	4.53	-32.70	3890.21	0.00	0.00	-0.64	0.64
86.00	1062.44	0.10	-2.29	4.22	-24.93	3597.03	0.00	0.00	-0.46	0.46
87.00	1062.40	0.09	-3.82	4.04	-18.21	3435.76	0.00	0.00	-0.33	0.33
88.00	1062.35	0.08	-4.88	3.95	-12.83	3348.34	0.00	0.00	-0.23	0.23
89.00	1062.31	0.06	-5.61	3.90	-8.78	3301.32	0.00	0.00	-0.15	0.15
90.00	1062.26	0.05	-6.09	3.88	-5.84	3276.02	0.00	0.00	-0.10	0.10
91.00	1062.22	0.04	-6.41	3.87	-3.79	3262.31	0.00	0.00	-0.07	0.07
92.00	1062.17	0.03	-6.61	3.87	-2.39	3254.79	0.00	0.00	-0.04	0.04
93.00	1062.13	0.02	-6.74	3.87	-1.46	3250.58	0.00	0.00	-0.03	0.03
94.00	1062.08	0.02	-6.82	3.87	-0.86	3248.19	0.00	0.00	-0.01	0.02
95.00	1062.04	0.01	-6.86	3.87	-0.48	3246.82	0.00	0.00	-0.01	0.01
96.00	1061.99	0.01	-6.88	3.87	-0.25	3246.03	0.00	0.00	0.00	0.00
97.00	1061.95	0.01	-6.90	3.87	-0.12	3245.57	0.00	0.00	0.00	0.00
98.00	1061.90	0.01	-6.90	3.87	-0.04	3245.29	0.00	0.00	0.00	0.00
99.00	1061.86	0.01	-6.90	3.87	-0.01	3245.11	0.00	0.00	0.00	0.00
100.00	1061.81	0.01	-6.90	3.87	0.01	3244.99	0.00	0.00	0.00	0.00
101.00	1061.77	0.01	-6.90	3.87	0.01	3244.88	0.00	0.00	0.00	0.00
102.00	1061.72	0.01	-6.90	3.87	0.01	3244.78	0.00	0.00	0.00	0.00
103.00	1061.68	0.01	-6.90	3.87	0.01	3244.68	0.00	0.00	0.00	0.00
104.00	1061.64	0.01	-6.90	3.87	0.01	3244.57	0.00	0.00	0.00	0.00
105.00	1061.59	0.01	-6.90	3.87	0.00	3244.46	0.00	0.00	0.00	0.00
106.00	1061.55	0.01	-6.90	3.87	0.00	3244.34	0.00	0.00	0.00	0.00
107.00	1061.50	0.01	-6.90	3.87	0.00	3244.23	0.00	0.00	0.00	0.00

Figure 8. (Continued)

108.00	1061.46	0.01	-6.90	3.87	0.00	3244.11	0.00	0.00	0.00	0.00
109.00	1061.41	0.01	-6.90	3.87	0.00	3243.99	0.00	0.00	0.00	0.00
110.00	1061.37	0.01	-6.90	3.87	0.00	3243.87	0.00	0.00	0.00	0.00
111.00	1061.32	0.01	-6.90	3.87	0.00	3243.74	0.00	0.00	0.00	0.00
112.00	1061.28	0.01	-6.90	3.87	0.00	3243.62	0.00	0.00	0.00	0.00
113.00	1061.23	0.01	-6.90	3.87	0.00	3243.50	0.00	0.00	0.00	0.00
114.00	1061.19	0.01	-6.90	3.87	0.00	3243.38	0.00	0.00	0.00	0.00
115.00	1061.14	0.01	-6.90	3.87	0.00	3243.26	0.00	0.00	0.00	0.00
116.00	1061.10	0.01	-6.90	3.87	0.00	3243.14	0.00	0.00	0.00	0.00
117.00	1061.06	0.01	-6.90	3.87	0.00	3243.02	0.00	0.00	0.00	0.00
118.00	1061.01	0.01	-6.90	3.87	0.00	3242.90	0.00	0.00	0.00	0.00
119.00	1060.97	0.01	-6.90	3.87	0.00	3242.78	0.00	0.00	0.00	0.00
120.00	1060.92	0.01	-6.90	3.87	0.00	3242.66	0.00	0.00	0.00	0.00
121.00	1060.88	0.01	-6.90	3.87	0.00	3242.54	0.00	0.00	0.00	0.00
122.00	1060.83	0.01	-6.90	3.87	0.00	3242.42	0.00	0.00	0.00	0.00
123.00	1060.79	0.01	-6.90	3.87	0.00	3242.30	0.00	0.00	0.00	0.00
124.00	1060.74	0.01	-6.90	3.87	0.00	3242.18	0.00	0.00	0.00	0.00
125.00	1060.70	0.01	-6.90	3.87	0.00	3242.06	0.00	0.00	0.00	0.00
126.00	1060.65	0.01	-6.90	3.87	0.00	3241.93	0.00	0.00	0.00	0.00
127.00	1060.61	0.01	-6.90	3.87	0.00	3241.81	0.00	0.00	0.00	0.00
128.00	1060.57	0.01	-6.90	3.87	0.00	3241.69	0.00	0.00	0.00	0.00
129.00	1060.52	0.01	-6.90	3.87	0.00	3241.57	0.00	0.00	0.00	0.00
130.00	1060.48	0.01	-6.90	3.87	0.00	3241.45	0.00	0.00	0.00	0.00
131.00	1060.43	0.01	-6.90	3.87	0.00	3241.33	0.00	0.00	0.00	0.00
132.00	1060.39	0.01	-6.90	3.87	0.00	3241.21	0.00	0.00	0.00	0.00
133.00	1060.34	0.01	-6.90	3.87	0.00	3241.09	0.00	0.00	0.00	0.00
134.00	1060.30	0.01	-6.90	3.87	0.00	3240.97	0.00	0.00	0.00	0.00
135.00	1060.25	0.01	-6.90	3.87	0.00	3240.85	0.00	0.00	0.00	0.00
136.00	1060.21	0.01	-6.90	3.87	0.00	3240.73	0.00	0.00	0.00	0.00
137.00	1060.16	0.01	-6.90	3.87	0.00	3240.61	0.00	0.00	0.00	0.00
138.00	1060.12	0.01	-6.90	3.87	0.00	3240.49	0.00	0.00	0.00	0.00
139.00	1060.07	0.01	-6.90	3.87	0.00	3240.37	0.00	0.00	0.00	0.00
140.00	1060.03	0.01	-6.90	3.87	0.00	3240.25	0.00	0.00	0.00	0.00
141.00	1059.99	0.01	-6.90	3.87	0.00	3240.13	0.00	0.00	0.00	0.00
142.00	1059.94	0.01	-6.90	3.87	0.00	3240.01	0.00	0.00	0.00	0.00
143.00	1059.90	0.01	-6.90	3.87	0.00	3239.89	0.00	0.00	0.00	0.00
144.00	1059.85	0.01	-6.90	3.87	0.00	3239.77	0.00	0.00	0.00	0.00
145.00	1059.81	0.01	-6.90	3.86	0.00	3239.65	0.00	0.00	0.00	0.00
146.00	1059.76	0.01	-6.90	3.86	0.00	3239.53	0.00	0.00	0.00	0.00
147.00	1059.72	0.01	-6.90	3.86	0.00	3239.41	0.00	0.00	0.00	0.00
148.00	1059.67	-0.11	-5.44	6.68	60.09	6087.32	0.00	-0.09	1.59	1.59
149.00	1059.60	-0.20	3.00	9.93	69.51	10586.41	0.00	-0.02	2.61	2.61
150.00	1059.49	-0.22	14.18	11.13	71.29	13647.36	0.00	-0.01	2.94	2.94
151.00	1059.38	-0.21	26.17	11.26	71.31	13931.10	0.00	0.00	2.97	2.97
152.00	1059.26	-0.19	38.02	10.70	70.30	12719.90	0.00	0.01	2.82	2.82
153.00	1059.16	-0.14	49.11	9.78	68.35	10123.20	0.00	0.01	2.55	2.55
154.00	1059.07	-0.09	59.02	8.72	65.35	6840.56	0.00	0.02	2.22	2.22
155.00	1059.00	-0.03	67.52	7.62	61.16	5697.24	0.00	0.02	1.85	1.85
156.00	1058.94	0.02	74.53	6.57	55.64	6153.72	0.00	0.02	1.49	1.49
157.00	1058.87	0.07	80.12	5.70	48.81	5412.48	0.00	0.01	1.16	1.16
158.00	1058.81	0.10	84.44	5.03	41.00	4468.51	0.00	0.01	0.88	0.88
159.00	1058.76	0.11	87.69	4.55	32.85	3923.03	0.00	0.01	0.65	0.65
160.00	1058.71	0.11	90.09	4.24	25.15	3616.97	0.00	0.00	0.47	0.47
161.00	1058.66	0.10	91.82	4.05	18.50	3446.90	0.00	0.00	0.33	0.33
162.00	1058.62	0.09	93.04	3.95	13.17	3352.98	0.00	0.00	0.23	0.23
163.00	1058.57	0.07	93.90	3.89	9.12	3301.15	0.00	0.00	0.16	0.16

Figure 8. (Continued)

164.00	1058.53	0.06	94.48	3.87	6.17	3272.37	0.00	0.00	0.11	0.11
165.00	1058.48	0.04	94.87	3.86	4.08	3256.19	0.00	0.00	0.07	0.07
166.00	1058.44	0.03	95.13	3.86	2.64	3246.93	0.00	0.00	0.05	0.05
167.00	1058.39	0.02	95.29	3.86	1.66	3241.52	0.00	0.00	0.03	0.03
168.00	1058.35	0.02	95.39	3.86	1.01	3238.30	0.00	0.00	0.02	0.02
169.00	1058.30	0.01	95.46	3.86	0.59	3236.36	0.00	0.00	0.01	0.01
170.00	1058.26	0.01	95.49	3.86	0.33	3235.18	0.00	0.00	0.01	0.01
171.00	1058.22	0.01	95.51	3.86	0.17	3234.46	0.00	0.00	0.00	0.00
172.00	1058.17	0.01	95.52	3.86	0.08	3234.02	0.00	0.00	0.00	0.00
173.00	1058.13	0.01	95.53	3.86	0.03	3233.74	0.00	0.00	0.00	0.00
174.00	1058.08	0.01	95.53	3.86	0.01	3233.55	0.00	0.00	0.00	0.00
175.00	1058.04	0.01	95.53	3.86	-0.01	3233.41	0.00	0.00	0.00	0.00
176.00	1057.99	0.01	95.53	3.86	-0.01	3233.29	0.00	0.00	0.00	0.00
177.00	1057.95	0.01	95.53	3.86	-0.01	3233.18	0.00	0.00	0.00	0.00
178.00	1057.90	0.01	95.53	3.86	-0.01	3233.07	0.00	0.00	0.00	0.00
179.00	1057.86	0.01	95.53	3.86	0.00	3232.96	0.00	0.00	0.00	0.00
180.00	1057.82	0.01	95.53	3.86	0.00	3232.85	0.00	0.00	0.00	0.00
181.00	1057.77	0.01	95.53	3.86	0.00	3232.74	0.00	0.00	0.00	0.00
182.00	1057.73	0.01	95.53	3.86	0.00	3232.62	0.00	0.00	0.00	0.00
183.00	1057.68	0.01	95.53	3.86	0.00	3232.50	0.00	0.00	0.00	0.00
184.00	1057.64	0.01	95.53	3.86	0.00	3232.38	0.00	0.00	0.00	0.00
185.00	1057.59	0.01	95.53	3.86	0.00	3232.26	0.00	0.00	0.00	0.00
186.00	1057.55	0.01	95.53	3.86	0.00	3232.14	0.00	0.00	0.00	0.00
187.00	1057.50	0.01	95.53	3.86	0.00	3232.02	0.00	0.00	0.00	0.00
188.00	1057.46	0.01	95.53	3.86	0.00	3231.91	0.00	0.00	0.00	0.00
189.00	1057.41	0.01	95.53	3.86	0.00	3231.79	0.00	0.00	0.00	0.00
190.00	1057.37	0.01	95.53	3.86	0.00	3231.67	0.00	0.00	0.00	0.00
191.00	1057.33	0.01	95.53	3.86	0.00	3231.55	0.00	0.00	0.00	0.00
192.00	1057.28	0.01	95.53	3.86	0.00	3231.43	0.00	0.00	0.00	0.00
193.00	1057.24	0.01	95.53	3.86	0.00	3231.31	0.00	0.00	0.00	0.00
194.00	1057.19	0.01	95.53	3.86	0.00	3231.19	0.00	0.00	0.00	0.00
195.00	1057.15	0.01	95.53	3.86	0.00	3231.07	0.00	0.00	0.00	0.00
196.00	1057.10	0.01	95.53	3.86	0.00	3230.95	0.00	0.00	0.00	0.00
197.00	1057.06	0.01	95.53	3.86	0.00	3230.83	0.00	0.00	0.00	0.00
198.00	1057.01	0.01	95.53	3.86	0.00	3230.71	0.00	0.00	0.00	0.00
199.00	1056.97	0.01	95.53	3.86	0.00	3230.59	0.00	0.00	0.00	0.00
200.00	1056.93	0.01	95.53	3.86	0.00	3230.47	0.00	0.00	0.00	0.00
201.00	1056.88	0.01	95.53	3.86	0.00	3230.35	0.00	0.00	0.00	0.00
202.00	1056.84	0.01	95.53	3.85	0.00	3230.23	0.00	0.00	0.00	0.00
203.00	1056.79	0.01	95.53	3.85	0.00	3230.11	0.00	0.00	0.00	0.00
204.00	1056.75	0.01	95.53	3.85	0.00	3229.99	0.00	0.00	0.00	0.00
205.00	1056.70	0.01	95.53	3.85	0.00	3229.88	0.00	0.00	0.00	0.00
206.00	1056.66	0.01	95.53	3.85	0.00	3229.76	0.00	0.00	0.00	0.00
207.00	1056.61	0.01	95.53	3.85	0.00	3229.64	0.00	0.00	0.00	0.00
208.00	1056.57	0.01	95.53	3.85	0.00	3229.52	0.00	0.00	0.00	0.00
209.00	1056.52	0.01	95.53	3.85	0.00	3229.40	0.00	0.00	0.00	0.00
210.00	1056.48	0.01	95.53	3.85	0.00	3229.28	0.00	0.00	0.00	0.00
211.00	1056.44	0.01	95.53	3.85	0.00	3229.16	0.00	0.00	0.00	0.00
212.00	1056.39	0.01	95.53	3.85	0.00	3229.04	0.00	0.00	0.00	0.00
213.00	1056.35	0.01	95.53	3.85	0.00	3228.92	0.00	0.00	0.00	0.00
214.00	1056.30	0.01	95.53	3.85	0.00	3228.80	0.00	0.00	0.00	0.00
215.00	1056.26	0.01	95.53	3.85	0.00	3228.68	0.00	0.00	0.00	0.00
216.00	1056.21	0.01	95.53	3.85	0.00	3228.56	0.00	0.00	0.00	0.00
217.00	1056.17	0.01	95.53	3.85	0.00	3228.44	0.00	0.00	0.00	0.00
218.00	1056.12	0.01	95.53	3.85	0.00	3228.33	0.00	0.00	0.00	0.00
219.00	1056.08	0.01	95.53	3.85	0.00	3228.21	0.00	0.00	0.00	0.00

Figure 8. (Continued)

220.00	1056.04	0.01	95.53	3.85	0.00	3228.09	0.00	0.00	0.00	0.00
221.00	1055.99	0.01	95.53	3.85	0.00	3227.97	0.00	0.00	0.00	0.00
222.00	1055.95	0.01	95.53	3.85	0.00	3227.85	0.00	0.00	0.00	0.00
223.00	1055.90	0.01	95.53	3.85	0.00	3227.73	0.00	0.00	0.00	0.00

Figure 8. (Continued)

CONCLUSIONS

This section briefly describes improvements to MOFO, certain tradeoffs between fidelity and run time, and future work. The following information supplements that in [3].

VARIABLE RANGE

At present, MOFO computes the trajectory of an aircraft while en route toward a specific waypoint in space. It determines that the waypoint has been achieved if the trajectory comes within a predetermined "basket." This basket is described as a set range from the waypoint. If the user either wishes to have more flexibility in this predetermined range or to have the aircraft come as close as possible to the waypoint, two suggestions are given.

If more flexibility in controlling the course of the trajectory is desired, a simple modification should suffice. For example, the user may input a specific range at the beginning of the command file or provide a range to accompany each WAYPT command.

Under some circumstances, however, it may be desirable to have the aircraft follow the desired trajectory more closely. In this instance, the basket essentially shrinks to a point. This situation is more difficult to model. This capability would depend on the severity of the maneuver and the current values of the velocity, DELTA, and DT. The mathematical development is left for future users.

OTHER CONTROL OPTIONS

Currently, MOFO has only one control option, WAYPT. Users may wish to have more flexibility in controlling the attitude and direction of the aircraft. The source code of subroutine CONTROL in appendix B contains other control routines that have been "commented" out. These routines are yet to be developed but provide a foundation for needed improvements. They are:

- HEADIN. This subroutine would allow control of the velocity heading angle, χ.
- PITCH. This subroutine would allow control of the velocity pitch angle, γ.
- HDGPTC. This subroutine would allow control of both the velocity heading and velocity pitch angle, χ and γ, respectively.
- DRPSTR. This subroutine would allow control of the dropping of external stores from the aircraft amounting to a weight reduction.
- TERFOL. This subroutine would allow the aircraft to perform terrain following to a waypoint. It would require the acquisition and manipulation of terrain elevation data.

Another control routine not included in the source code will also need to be developed if necessity dictates. Namely, MOFO reads in thrust and fuel flow data for both military and afterburner power. However, the model computes thrust under military power

only. A switch would need to be implemented if the user wishes to fly the aircraft in afterburner power.

FIDELITY VERSUS CPU TIME

As with all models, MOFO is not without the tradeoff that exists between accuracy and run time. In MOFO, a reasonable compromise between fidelity and simplicity has been achieved in bringing the model to its current one-to-one ratio of actual-to-simulated time. For instance, MOFO does not model the aerodynamic response at the control surface level (simulating the response to deflections of the rudder, fins, ailerons, etc., as described in [4]). This level of detail was not required for the intended use of the model.

If a shorter run time is required with position and attitude information at a lesser degree of accuracy, some suggestions are in order. One approach would be to convert the double precision format into single precision. Another would be to use linear interpolation as opposed to cubic and bicubic spline interpolation. A third possibility would be an integration algorithm other than Runge-Kutta. Although the Runge-Kutta method is self-starting, a great deal of computer time is required for the four intermediate function evaluation steps. An alternative may be to use the Adams formulas (such as those found in [5]) after the first few Runge-Kutta iterations.

CURRENT WORK

At the time of this writing, work is under way both to validate and modify MOFO. The validation effort has involved the comparison of performance predictions from MOFO and the generally accepted flight path generator, BLUEMAX II, to aerodynamic charts found in NATOPS manuals. MOFO is being modified to include implementing a variable DELTA time step that depends on the maneuvers being performed by the aircraft (see [6]).

REFERENCES

[1] CNA Research Memorandum 89-63, *An Analyst's Guide to the MOFO Model*, by Michael W. Price and James E. Schliessmann, Unclassified, May 1989 (27890063)[1]

[2] CNA Research Memorandum 89-93, *GEMM88 Analyst's Manual*, by Stuart G. Dunn, Unclassified, forthcoming (27890093)

[3] Chuan-Tau Edward Lan and Jan Roskam. *Airplane Aerodynamics and Performance.* Kansas: Roskam Aviation and Engineering Co., 1980

[4] Bernard Etkin. *Dynamics of Atmospheric Flight.* New York: John Wiley & Sons, Inc., 1972

[5] Robert W. Hornbeck. *Numerical Methods.* New Jersey: Quantum Publishers, 1975

[6] CNA Research Memorandum 89-96, *A Comparison of MOFO and BLUEMAX Generated Turn Measures for the F/A-18*, by A. F. Barghouty, Confidential, forthcoming (27890096)

1. The numbers in parentheses are CNA internal control numbers.

APPENDIX A

MODULE AND VARIABLE DESCRIPTIONS

APPENDIX A

MODULE AND VARIABLE DESCRIPTIONS

PROGRAM MOFO

The program MOFO is the main module of the MOFO model. MOFO begins by initializing several variables that are used throughout the rest of the program. It then reads in the aircraft input data file. This file consists of one-dimensional and two-dimensional tables. Program MOFO reads in the one-dimensional tables and calls subroutine TABIN to read the two-dimensional tables. Subroutines SPLINE and SPLIE2 are called concurrently to create second derivative tables that will be used later when cubic and bicubic spline interpolation is required. For some of the tables, subroutine PARDER is called to create partial derivative tables to be used later in the root-finding subroutine, NWTN1D. Finally, program MOFO calls subroutine CONTROL to begin processing the command file.

MOFO Common Variables

The following are variables placed in common blocks that are used throughout the model MOFO. These common blocks are located in the file called CMNBLK.CMN. Most routines have an "INCLUDE" statement to incorporate this structure.

Parameters

NMAX—parameter descriptive of the maximum array size currently set to 14

Common Block /AERO/

AIRSPD(NMAX)—array of Mach numbers of length NAOAMX

AOAMAX(NMAX)—array of the maximum angle of attack for each Mach number in AIRSPD(NMAX) (degrees)

AAMXSD(NMAX)—array that contains the second derivatives of AOAMAX(NMAX) at each Mach number in AIRSPD(NMAX)

CLFTCD(NMAX)—array of lift coefficients of length NCLCD

MACHCD(NMAX)—array of Mach numbers of length NMCHCD

CD—interpolated value of the drag coefficient, C_D

CDRAG(NMAX,NMAX)—two-dimensional array of the drag coefficients as a function of CLFTCD(NMAX) and MACHCD(NMAX), of size NCLCD by NMCHCD

CDRGSD(NMAX,NMAX)—two-dimensional array that contains the second derivatives of the drag coefficients associated with each value in CDGRAG(NMAX,NMAX)

AOACL(NMAX)—array of angles of attack of length NAOACL (degrees)

MACHCL(NMAX)—array of Mach numbers of length NMCHCL

CL—interpolated value of the lift coefficient, C_L

CLIFT(NMAX,NMAX)—two-dimensional array of the lift coefficients as a function of AOACL(NMAX) and MACHCL(NMAX), of size NAOACL by NMCHCL

CLFTSD(NMAX,NMAX)—two-dimensional array that contains the second derivatives of the lift coefficients associated with each value of CLIFT(NMAX,NMAX)

DERCD—interpolated value of the partial derivative, $\partial C_D/\partial C_L$

DCDDCL(NMAX,NMAX)—two-dimensional array of the partial derivative, $\partial C_D/\partial C_L$, of size NCLCD and NMCHCD associated with each value of CDRAG(NMAX,NMAX)

DCDDCLSD(NMAX,NMAX)—two-dimensional array that contains the second derivatives of each value in DCDDCL(NMAX,NMAX)

DERCL—interpolated value of the partial derivative, $\partial C_L/\partial \alpha$

DCLDA(NMAX,NMAX)—two-dimensional array of the partial derivative, $\partial C_L/\partial \alpha$, associated with each value of CLIFT(NMAX,NMAX)

DCLDASD(NMAX,NMAX)--two-dimensional array that contains the second derivatives of each value in DCLDA(NMAX,NMAX)

Common Block /AIRCFT/

ACM—total mass of aircraft, fuel, and armaments (slugs)

AREA—planform area (ft^2)

BINGO—amount of fuel left that signals the pilot to return to base (lb)

WTAC—base weight of the aircraft (lb)

WTFLMX—maximum weight of fuel that can be loaded on the planform (lb)

WTFUEL—initial weight of fuel placed in the command file (lb)

Common Block /AOA/

ALPHA—present angle of attack after time filtering

ALPHA0—initial or previous angle of attack

AOATAU—aerodynamic time constant for angle of attack filtering (seconds)

Common Block /CONSTS/

G—acceleration due to gravity (32.17 ft/sec^2)

COUNT—value of 1/DT

ICOUNT—integer value of COUNT

MTOFT—a conversion factor: meters to feet (3.280840 ft/m)

P0—reference pressure at sea level (2116.2 lb/ft^2)

PI—π (3.14159265358979)

RHO0—density of atmosphere at sea level (2.377 x 10^{-3} slugs/ft^3)

ROOT—value of 1/3.5 used in the RPM calculation in routine AERO

RTODEG—a conversion factor: radians to degrees (180/π degrees/radian)

Common Block /CONTRL/

CHIDOT—present commanded heading rate after time filtering

CHIDOTO—previous commanded heading rate

CHITAU—time constant for filtering yaw rate commands (seconds)

GMDOT—present commanded pitch rate after time filtering

GMDOT0—previous commanded pitch rate

GMTAU—time constant for filtering pitch rate commands (seconds)

VDOT—present commanded acceleration after time filtering

VDOT0—previous commanded acceleration

VTAU—time constant for filtering longitudinal accleration commands (seconds)

DELTA—control time step interval between acceleration commands (seconds)

DT—differential time step for the Runge-Kutta algorithm (seconds)

GEES—magnitude of the load factor that the platform experiences

KAY—constant for proportional navigation

Common Block /CYCLE/

MV—momentum (mass times acceleration due to velocity)

MG—weight (mass times gravity)

REFOR—reference force ($.5*\rho*V^2*AREA$)

SNG—sin(γ); sine of the velocity pitch angle

CSG—cos(γ); cosine of the velocity pitch angle

SNCHI—sin (χ); sine of the velocity heading angle

CSCHI—cos(χ); cosine of the velocity heading angle

CT—interpolated value of the thrust specific fuel consumption

Common Block /FNCTN/

F1—$\dot{X}$; the velocity component in the X direction

F2—$\dot{Y}$; the velocity component in the Y direction

F3—$\dot{Z}$; the velocity component in the Z direction

F4—$\dot{V}$; the longitudinal acceleration component along the velocity vector

F5—$\dot{\gamma}$; the pitch rate

F6—$\dot{\chi}$; the yaw rate

F7—$\dot{m}$; the rate of change of the aircraft's mass

Common Block /POWER/

ALT(NMAX)—array of altitudes of length NMCHMX (feet)

SPDMAX(NMAX)—array of maximum Mach numbers for each altitude in ALT(NMAX)

SPMXSD(NMAX)—array of second derivatives associated with each value in SPDMAX(NMAX)

ALTMIL(NMAX)—array of altitudes of length NALTML (feet)

MCHMIL(NMAX)—array of Mach numbers of length NMCHML

PWRMIL(NMAX,NMAX)—two-dimensional array of maximum thrust under military power as a function of ALTMIL(NMAX) and MCHMIL(NMAX) of size NALTML by NMCHML (lb)

PRMLSD(NMAX,NMAX)—two-dimensional array of the second derivatives associated with each value in PWRMIL(NMAX,NMAX)

FULMIL(NMAX,NMAX)—two-dimensional array of maximum fuel flow rates under military power as a function of ALTMIL(NMAX) and MCHMIL(NMAX) of size NALTML by NMCHML (lb/hr)

FLMLSD(NMAX,NMAX)—two-dimensional array of the second derivatives associated with each value in FULMIL(NMAX,NMAX)

ALTAB(NMAX)—array of altitudes of length NALTAB (feet)

MCHAB(NMAX)—array of Mach numbers of length NMCHAB

PWRAB(NMAX,NMAX)—two-dimensional array of maximum thrust under afterburner power as a function of ALTAB(NMAX) and MCHAB(NMAX) of size NALTAB by NMCHAB (lb)

PRABSD(NMAX,NMAX)—two-dimensional array of the second derivatives associated with each value in PWRAB(NMAX,NMAX)

FULAB(NMAX,NMAX)—two-dimensional array of fuel flow rates under afterburner power as a function of ALTAB(NMAX) and MCHAB(NMAX) of size NALTAB by NMCHAB (lb/hr)

FLABSD(NMAX,NMAX)—two-dimensional array of the second derivatives associated with each value in FULAB(NMAX,NMAX)

PCTRPM(NMAX)—array of percentages of maximum revolutions per minute (RPMs) of the turbojet engine of length NCTRPM

TSFC(NMAX)—array of thrust specific fuel consumption for each value in PCTRPM(NMAX)

TSFCSD(NMAX)—array of the second derivatives associated with each value in TSFC(NMAX) (lb/hr/lb)

MACH—Mach number at the aircraft's present altitude

TH—computed thrust value needed to satisfy the equations of motion

THMAX—interpolated value of the maximum thrust at the aircraft's present altitude and Mach number

Common Block /ROLL/

MU—roll angle after time filtering

MU0—initial or previous roll angle

MUTAU—aerodynamic time constant for roll angle filtering (seconds)

Common Block /TBLMAX/

NMCHMX—the number of values in ALT(NMAX), SPDMAX(NMAX), and SPMXSD(NMAX)

NAOAMX—the number of values in AIRSPD(NMAX), AOAMAX(NMAX), and AAMXSD(NMAX)

NALTML—the number of values in ALTMIL(NMAX)

NMCHML—the number of values in MCHMIL(NMAX)

NALTAB—the number of values in ALTAB(NMAX)

NMCHAB—the number of values in MCHAB(NMAX)

NCLCD—the number of values in CLFTCD(NMAX)

NMCHCD—the number of values in MACHCD(NMAX)

NAOACL—the number of values in AOACL(NMAX)

NMCHCL—the number of values in MACHCL(NMAX)

NCTRPM—the number of values in PCTRPM(NMAX), TSFC(NMAX), and TSFCSD(NMAX)

MOFO Local Variables

The following variables are intrinsic only to program MOFO:

I—counter for DO loop

IALT(NMAX)—array of altitudes in integer form that is later converted to double precision real and placed in array ALT(NMAX) (feet).

SUBROUTINE AERO

The subroutine AERO is called by the various control routines. Its function is to compute the aerodynamic forces based on the commanded accelerations. Required values of the angle of attack, the velocity roll angle, and thrust are determined via a root-finding algorithm in subroutine NWTN1D. The passed parameters are:

X—the present value of the X coordinate

Y—the present value of the Y coordinate

Z—the present value of the Z coordinate

V—the present magnitude of the velocity vector

GM—the present velocity pitch angle, γ

CHI—the present velocity yaw angle, χ

T—the present time on the model's clock.

AERO's local variables are:

H—the altitude of the aircraft

ROH—the atmospheric density, ρ, at altitude H

VSQ—the square of the present speed, V^2

MUI—the instructed roll angle, μ_I, calculated from the directional accelerations, CHIDOT and GMDOT

ALPHAI—the instructed angle of attack, α_I, returned from subroutine NWTN1D; the present ALPHA is used as an initial guess

TEMALF—a temporary place holder when limiting ALPHAI between $-\pi$ and π

ALFMAX—the maximum angle of attack at the present Mach number

TH—the present thrust needed to satisfy the commanded aerodynamic conditions

THMAX—the maximum thrust at the present altitude and Mach number

RPM—the percentage of maximum revolutions per minute (RPMs) of the turbojet engine as a function of TH and THMAX.

SUBROUTINE ATTITUDE

The subroutine ATTITUDE is called by subroutine CONTROL and subroutine RK. At present, this subroutine is called once every second of simulation time. It determines the body attitude of the aircraft and prints the results into a file. The passed parameters are:

X1—simulation time

X2—X

X3—Y

X4—Z

X5—V

X6—γ

X7—χ

X8—α

X9—μ.

ATTITUDE's local variables are:

CAL—cosine of α

SAL—sine of α

CGM—cosine of γ

SGM—sine of γ

CCH—cosine of χ

SCH—sine of χ

CMU—cosine of μ

SMU—sine of μ

AA—X-component of the $\hat{i}_b$ unit vector in the internal local horizon system

BB—Y-component of the $\hat{i}_b$ unit vector in the internal local horizon system

CC—Z-component of the $\hat{i}_b$ unit vector in the internal local horizon system

BCHI—the body yaw angle with respect to the local horizon

BGM—the body pitch angle with respect to the local horizon.

SUBROUTINE CONTROL

The subroutine CONTROL is called from program MOFO. Its task it to process the command file. First, the initial conditions of the aircraft are read in. Then subsequent

orders are read in and transferred to the appropriate subroutine to complete the command. The subroutine's local variables are:

X0—initial X position of aircraft (m)

Y0—initial Y position of aircraft (m)

Z0—initial Z position of aircraft (m)

V0—initial speed of aircraft (m/sec)

GM0—initial velocity pitch angle of aircraft (degrees)

CHI0—initial velocity yaw angle of aircraft (degrees)

T0—starting time (seconds)

WTARM—weight of external stores carried on aircraft (lb)

X—X0 value in the internal coordinate system (feet)

Y—Y0 value in the internal coordinate system (feet)

Z—Z0 value in the internal coordinate system (feet)

V—V0 value in the internal coordinate system (ft/sec)

GM—GM0 value in the internal coordinte system (radians)

CHI—CHI0 value in the internal coordinate system (radians)

T—T0 value from the command input file (seconds).

SUBROUTINE F

The subroutine F computes the instantaneous time rates of change of an aircraft's Cartesian position, speed, pitch and yaw angles, and mass. Subroutine F is called by subroutine RK four times to integrate these time rates over the interval DT. The passed parameters are:

X—the X value at one of four integration steps

Y—the Y value at one of four integration steps

Z—the Z value at one of four integration steps

V—the V value at one of four integration steps

GM—the γ value at one of four integration steps

CHI—the χ value at one of four integration steps

MAC—the value of the aircraft's mass, ACM, at one of four integration steps

T—the present simulated time

F's local variables are:

H—altitude of aircraft

ROH—the atmospheric density ρ at altitude H

VSQ—square of present velocity, V^2

ALNG—longitudinal acceleration

APTCH—pitch acceleration

AYW—yaw acceleration

ATOTAL—magnitude of the total acceleration.

FUNCTION FALPHA

The function FALPHA is called by subroutine AERO. This function filters the angle of attack. A first-order linear time filter is used in the calculation. This filter is discussed in detail in [A-1]. The passed parameter is:

ALPHAI—commanded angle of attack, α_I.

FALPHA's local variables are:

TERM—the exponential $e^{-DT/AOATAU}$

FALPHA—function value returned.

FUNCTION FCHIDOT

The function FCHIDOT is called from the various control routines. This function filters the yaw rate. A first-order linear time filter is used in the calculation. This filter is discussed in detail in [A-1]. The passed parameter is:

CHIDOT—commanded yaw rate, $\dot{\chi}_I$.

FCHIDOT's local variables are:

TERM—the exponential $e^{-DELTA/CHITAU}$

FCHIDOT—function value returned.

FUNCTION FDF1X

The function FDF1X is called by subroutine NWTN1D. It computes the derivative of the equation found in function FUNCF1 with respect to the independent variable. In this case, the independent variable is the instructed angle of attack, α_I. The passed parameter is:

ALPHAI—the instructed angle of attack, α_I.

FDFIX's local variable is:

FDF1X—function value returned.

FUNCTION FGMDOT

The function FGMDOT is called from the various control subroutines. This function filters the pitch rate. A first-order linear time filter is used in the calculation. This filter is discussed in detail in [A-1]. The passed parameter is:

GMDOTI—the commanded pitch velocity, $\dot{\gamma}_I$.

FGMDOT's local variables are:

TERM—the exponential $e^{-DELTA/GMTAU}$

FGMDOT—function value returned.

FUNCTION FMU

The function FMU is called from subroutine AERO. This function filters the roll angle. A first-order linear time filter is used in the calculation. This filter is discussed in detail in [A-1]. The passed parameter is:

MUI—commanded velocity roll angle, μ_I.

FMU's local variables are:

TERM—the exponential $e^{-DT/MUTAU}$

FMU—function value returned.

FUNCTION FUNCF1

The function FUNCF1 is called by the subroutine NWTN1D. This function calculates the difference between the commanded pitch rate and the computed pitch rate based on the aircraft's present aerodynamic conditions. Subroutine NWTN1D uses this

equation to estimate a value of the instructed angle of attack, α_I, that will minimize the difference. The passed parameter is:

ALPHAI—the passed X value from NWTN1D, which is the instructed angle of attack, α_I

FUNCF1's local variable is:

FUNCF1—function value returned.

FUNCTION FVDOT

The function FVDOT is called from the various control routines. This function filters the longitudinal acceleration. A first-order linear time filter is used in the calculation. This filter is discussed in detail in [A-1]. The passed parameter is:

VDOTI—the commanded longitudinal acceleration, $\dot{V}_I$

FUDOT's local variables are:

TERM—the exponential $e^{-DELTA/VTAU}$

FVDOT—function value returned.

SUBROUTINE NWTN1D

The subroutine NWTN1D is called from subroutine AERO. NWTN1D performs a one-dimensional root-finding algorithm using the Newton-Raphson method. This algorithm can be found in [A-2]. The passed parameter is:

X—the passed parameter from subroutine AERO; in this case, X is equal to the instructed angle of attack, α_I.

NWTN1D's local variables are:

FLAG—logical variable for the DO WHILE loop

EPSLNX—degree of tolerance for the difference between the present value of X and the value from the most recent iteration of the root-finding algorithm, XNEW; once this difference is less than the value of EPSLNX, the DO WHILE loop ends and the value of X is returned to AERO

JCOUNT—an integer counter keeping track of the number of iterations that the Newton-Raphson method is taking to satisfy the tolerance EPSLNX

FUNC1—function value returned from FUNCF1

DF1X—function value returned from FDF1X

DET—the determinant of the matrix derived from the system of equations being used in the root-finding algorithm; in the one-dimensional case, DET is equal to DF1X

DETINV—the inverse of the determinent, 1/DET

XNEW—the computed value of the root.

FUNCTION P

The function P computes the standard pressure as a function of elevation, H. The passed parameter is:

H—altitude above sea level (feet).

P's local variable is:

P—pressure at altitude H (lb/ft^2).

SUBROUTINE PARDER

The subroutine PARDER is called from program MOFO. A two-dimensional table ARRAY and an independent variable array X are received. Subroutine PARDER computes partial derivatives at each position in ARRAY with respect to the independent variable array X. The method used is the central-difference formula, except at the endpoints. The forward-difference formula is performed on the first endpoint while the backward-difference formula is performed on the last element in the array. The passed parameters are:

ARRAY(NMAX,NMAX)—the two-dimensional table that is passed to PARDER

DERIV(NMAX,NMAX)—the returned two-dimensional table of partial derivatives of ARRAY(NMAX,NMAX) with respect to X(NMAX)

X(NMAX)—the one-dimensional independent variable array that the derivative will be taken with respect to

IMAX—the maximum number of values in the array X(NMAX), also corresponds to the number of rows in ARRAY(NMAX,NMAX)

JMAX—the number of columns of ARRAY(NMAX,NMAX).

PARDER's local variables are:

NMAX—parameter descriptive of the maximum array size currently set to 14

I—row index for the DO loop

J—column index for the DO loop.

FUNCTION RHO

The function RHO computes the standard atmospheric density ρ as a function of elevation, H. The passed parameter is:

H—altitude above sea level (feet).

RHO's local variable is:

RHO—atmospheric density as a function of height (slugs/ft^3).

SUBROUTINE RK

The subroutine RK performs fourth-order Runge-Kutta integration of a system of first-order nonlinear differential equations (i.e., the equations of motion). The algorithm uses a time step DT. Subroutine F is called four times to complete the integration. See [A-1] and [A-2] for a more detailed explanation on Runge-Kutta integration. The passed parameters are:

X—the present X coordinate of the aircraft passed from a control routine; at the end of RK, X is updated over the time interval DT

Y—the present Y coordinate of the aircraft passed from a control routine; at the end of RK, Y is updated over the time interval DT

Z—the present Z coordinate of the aircraft passed from a control routine; at the end of RK, Z is updated over the time interval DT

V—the present magnitude of the velocity vector passed from a control routine; at the end of RK, V is updated over the time interval DT

GM—the present pitch angle of the velocity vector passed from a control routine; at the end of RK, GM is updated over the time interval DT

CHI—the present yaw angle of the velocity vector passed from a control routine; at the end of RK, CHI is updated over the time interval DT

T—the present time of the simulation passed from a control routine; at the end of RK, T is updated over the time interval DT.

RK's local variables are:

F11—$\dot{X}$; the velocity component in the X direction after the first call to subroutine F

F21—$\dot{Y}$; the velocity component in the Y direction after the first call to subroutine F

F31—$\dot{Z}$; the velocity component in the Z direction after the first call to subroutine F

F41—$\dot{V}$; the acceleration component along the velocity vector after the first call to subroutine F

F51—$\dot{\gamma}$; the pitch rate after the first call to subroutine F

F61—$\dot{\chi}$; the yaw rate after the first call to subroutine F

F71—$\dot{m}$; the rate of change in the aircraft's mass after the first call to subroutine F

XSIN—value as a function of X, DT, and F11 which is passed to subroutine F during the second call

YSIN—value as a function of Y, DT, and F21, which is passed to subroutine F during the second call

ZSIN—value as a function of Z, DT, and F31, which is passed to subroutine F during the second call

VSIN—value as a function of V, DT, and F41, which is passed to subroutine F during the second call.

GMSIN—value as a function of GM, DT, and F51, which is passed to subroutine F during the second call

CHSIN—value as a function of CHI,DT, and F61, which is passed to subroutine F during the second call

ACMSIN—value as a function of ACM, DT, and F71, which is passed to subroutine F during the second call

F12—$\dot{X}$; the velocity component in the X direction after the second call to subroutine F

F22—$\dot{Y}$; the velocity component in the Y direction after the second call to subroutine F

F32—$\dot{Z}$; the velocity component in the Z direction after the second call to subroutine F

F42—$\dot{V}$; the acceleration component along the velocity vector after the second call to subroutine F

F52—$\dot{\gamma}$; the pitch rate after the second call to subroutine F

F62—$\dot{\chi}$; the yaw rate after the second call to subroutine F

F72—$\dot{m}$; the rate of change in the aircraft's mass after the second call to subroutine F

XSSIN—value as a function of X, DT, and F12 which is passed to subroutine F during the third call

YSSIN—value as a function of Y, DT, and F22, which is passed to subroutine F during the third call

ZSSIN—value as a function of Z, DT, and F32, which is passed to subroutine F during the third call

VSSIN—value as a function of V, DT, and F42, which is passed to subroutine F during the third call

GMSSIN—value as a function of GM, DT, and F52, which is passed to subroutine F during the third call

CHSSIN—value as a function of CHI, DT, and F62, which is passed to subroutine F during the third call

ACMSSIN—value as a function of ACM, DT, and F72, which is passed to subroutine F during the third call

F13—$\dot{X}$; the velocity component in the X direction after the third call to subroutine F

F23—$\dot{Y}$; the velocity component in the Y direction after the third call to subroutine F

F33—$\dot{Z}$; the velocity component in the Z direction after the third call to subroutine F

F43—$\dot{V}$; the acceleration component along the velocity vector after the third call to subroutine F

F53—$\dot{\gamma}$; the pitch rate after the third call to subroutine F

F63—$\dot{\chi}$; the yaw rate after the third call to subroutine F

F73—$\dot{m}$; the rate of change in the aircraft's mass after the third call to subroutine F

XSP1—value as a function of X, DT, and F13, which is passed to subroutine F during the fourth call

YSP1—value as a function of Y, DT, and F23, which is passed to subroutine F during the fourth call

ZSP1—value as a function of Z, DT, and F33, which is passed to subroutine F during the fourth call

VSP1—value as a function of V, DT, and F43, which is passed to subroutine F during the fourth call

GMSP1—value as a function of GM, DT, and F53, which is passed to subroutine F during the fourth call

CHSP1—value as a function of CHI, DT, and F63, which is passed to subroutine F during the fourth call

ACMSP1—value as a function of ACM, DT, and F73, which is passed to subroutine F during the fourth call

F14—$\dot{X}$; the velocity component in the X direction after the fourth call to subroutine F

F24—$\dot{Y}$; the velocity component in the Y direction after the fourth call to subroutine F

F34—$\dot{Z}$; the velocity component in the Z direction after the fourth call to subroutine F

F44—$\dot{V}$; the acceleration component along the velocity vector after the fourth call to subroutine F

F54—$\dot{\gamma}$; the pitch rate after the fourth call to subroutine F

F64—$\dot{\chi}$; the yaw rate after the fourth call to subroutine F

F74—$\dot{m}$; the rate of change in the aircraft's mass after the fourth call to subroutine F.

SUBROUTINE SPLIE2

The subroutine SPLIE2 is called from program MOFO. Given an M by N tabulated function YA, and tabulated independent variables X1A (M values) and X2A (N values), this routine constructs one-dimensional natural cubic splines of the rows of YA and returns the second derivatives in the array Y2A. The FORTRAN code was extracted from [A-3]. The passed parameters are:

X1A(NMAX)—the passed one-dimensional array containing independent variables of size M

X2A(NMAX)—the passed one-dimensional array containing independent variables of size N

YA(NMAX,NMAX)—the passed two-dimensional table containing the tabulated function values of X1A(NMAX) and X2A(NMAX) of size M by N

M—integer describing the actual size of X1A(NMAX)

N—integer describing the actual size of X2A(NMAX)

Y2A(NMAX,NMAX)—the returned two-dimensional table containing the second-derivatives of the tabulated function values in YA(NMAX,NMAX).

SPLIE2's local variables are:

NMAX—parameter descriptive of the maximum array size currently set to 14

YTMP(NMAX)—one-dimensional array to store all the values of a row of YA(NMAX,NMAX)

Y2TMP(NMAX)—the second derivatives for each value in YTMP(NMAX) returned when subroutine SPLINE is called

J—counter for DO loop and used as an array index term

K—counter for DO loop and used as an array index term.

SUBROUTINE SPLIN2

The subroutine SPLIN2 is called by various routines. Given X1A, X2A, YA, M, N as described in SPLIE2 and Y2A as produced by that routine; and given a desired interpolating point X1, X2; this routine returns an interpolated function value Y by bicubic spline interpolation. The FORTRAN code was extracted from [A-3]. The passed parameters are:

X1A(NMAX)—the array as described in subroutine SPLIE2

X2A(NMAX)—the array as described in subroutine SPLIE2

YA(NMAX,NMAX)—the table as described in subroutine SPLIE2

Y2A(NMAX,NMAX)—the table as described in subroutine SPLIE2

M—integer as described in subroutine SPLIE2

N—integer as described in subroutine SPLIE2

X1—desired interpolating point for X1A(NMAX)

X2—desired interpolating point for X2A(NMAX)

Y—the returned interpolated function value.

SPLIN2's local variables are:

NMAX—parameter descriptive of the maximum array size currently set to 14

YTMP(NMAX)—one-dimensional array to store all the values of a row of YA(NMAX,NMAX)

Y2TMP(NMAX)—one-dimensional array to store all the values of a row of Y2A(NMAX,NMAX), then used to store the second derivatives of the array YYTMP(NMAX) when SPLINE is called

YYTMP(NMAX)—one-dimensional array created by interpolating YA(NMAX,NMAX) at X2 for all the rows

J—counter for DO loop and used as an array index term

K—counter for DO loop and used as an array index term.

SUBROUTINE SPLINE

The subroutine SPLINE is called by program MOFO. Given arrays X and Y of length N containing a tabulated function, and given value YP1 and YPN for the first derivative of the interpolating function at points 1 and N, respectively, this routine returns an array Y2 of length N, which contains the second derivatives of the interpolating function at the tabulated points X_i. If YP1 and/or YPN are equal to 1×10^{30} or larger, the routine is signalled to set the corresponding boundary condition for a natural spline, with zero second derivative on that boundary. The FORTRAN code was extracted from [A-3]. The passed parameters are:

X(NMAX)—the passed one-dimensional array containing the tabulated independent variables of size N

Y(NMAX)—the passed one-dimensional array containing the tabulated independent function values of size N

N—integer describing the actual size of X(NMAX) and Y(NMAX)

YP1—first derivative of the interpolating function at point 1

YPN—first derivative of the interpolating function at point N

Y2(NMAX)—one-dimensional array containing the second derivatives of the interpolating function at the tabulated points X_i.

SPLINE's local variables are:

NMAX—parameter descriptive of the maximum array size currently set to 14

U(NMAX)—an array that acts as an intermediate step in the calculation of the second derivative

I—counter for DO loop and used as an array index term

K—counter for DO loop and used as an array index term.

SUBROUTINE SPLINT

The subroutine SPLINT is called by various subroutines. Given the arrays XA and YA of length N, which tabulate a function (with the XA_is in order), given the array Y2A, which is the output from SPLINE above, and given a value of X, this routine returns a cubic-spline interpolated value Y. The FORTRAN code was extracted from [A-3]. The passed parameters are:

XA(NMAX)—the array as described in subroutine SPLINE

YA(NMAX)—the array as described in subroutine SPLINE

Y2A(NMAX—the array as described in subroutine SPLINE

N—integer as described in subroutine SPLINE

X—desired interpolating point of the array XA(NMAX)

Y—the returned interpolated function value.

SPLINT's local variables are:

NMAX—parameter descriptive of the maximum array size currently set to 14

KLO—index value to find the largest value of XA(NMAX) that still satisfies the relationship, XA(KLO) ≤ X

KHI—index value to find the smallest value of XA(NMAX) that still satisfies the relationship, XA(KHI) ≥ X

K—index value to locate KHI and KLO

H—difference between XA(KHI) and XA(KLO)

A—the ratio: [XA(KHI) - X]/H

B—the ratio: [X - XA(KLO)]/H.

SUBROUTINE TABIN

The subroutine TABIN is called from program MOFO to read in the power and aerodynamic tables from the aircraft characteristic file. A flag variable is passed to indicate whether the file is composed of real or integer values. Integer inputs are converted to real. Variable SCALE multiplies the table values, if not equal to one. The passed parameters are:

X1ARGS(NMAX)—the independent array values associated with each of the rows of the two-dimensional table

X2ARGS(NMAX)—the independent array values associated with each of the columns of the two-dimensional table

TABLE(NMAX,NMAX)—table to be read in

NROWS—number of rows in table

NCOLS—number of columns in table

KFLAG—flag that indicates whether the table and the first argument is composed of integer or real values.

TABIN's local variables are:

NMAX--parameter descriptive of the maximum array size currently set to 14

IARGS(NMAX)—integer form of the X1ARGS(NMAX) array

ITABLE(NMAX,NMAX)—integer table to be read in

SCALE—a scale factor that multiplies the table

IR—counter for number of rows in the DO loop

IC—counter for number of columns in the DO loop.

SUBROUTINE WAYPT

The subroutine WAYPT is called by subroutine CONTROL to construct a flight segment. The present position and velocity vector of the aircraft are passed in this call statement. The subroutine then reads in its commanded position and calculates the directional change necessary to point the velocity vector to the desired waypoint. Finally, it constructs a flight path by calling subroutine RK to achieve the desired position. The passed parameters are:

X—the present X coordinate of the aircraft

Y—the present Y coordinate of the aircraft

Z—the present Z coordinate of the aircraft

V—the present magnitude of the velocity of the aircraft

GM—the present velocity pitch angle

CHI—the present velocity yaw angle

T—the present simulated time of the model.

WAYPT's local variables are:

FLAG—logical for the DO loop

XF—X value of desired waypoint

ZF—Z value of desired waypoint

VF—V value of desired waypoint

DELX—difference between present X value and desired X value

DELY—difference between present Y value and desired Y value

DELZ—difference between present Z value and desired Z value

VX—the component of the velocity vector in the X direction

VY—the component of the velocity vector in the Y direction

VZ—the component of the velocity vector in the Z direction

RANGE—slant range between present position and desired waypoint

OMGALOSX—Ω_{LOS_X}; see [A-1] for details

OMGALOSY—Ω_{LOS_Y}; see [A-1] for details

OMGALOSZ—Ω_{LOS_Z}; see [A-1] for details

DELV—the difference between VF and V

AVEV—average speed of VF and V

TTI—time to intercept calculated by RANGE and AVEV

DRAGACCEL—maximum acceleration due to drag

RATIO—variable used as a check to be sure that the commanded directional changes are not greater than the user-imposed g-limits; if they are, they are scaled accordingly

CHIDOTI—the computed yaw rate command before filtering, $\dot{\chi}_I$

GMDOTI—the computed pitch rate command before filtering, $\dot{\gamma}_I$

VDOTI—the computed longitudinal acceleration command before filtering, $\dot{V}_I$

TF—simulated time when the acceleration commands are updated.

REFERENCES

[A-1] CNA Research Memorandum 89-63, *An Analyst's Guide to the MOFO Model*, by Michael W. Price and James E. Schliessmann, Unclassified, May 1989 (27890063)[1]

[A-2] Robert W. Hornbeck. *Numerical Methods*. New Jersey: Quantum Publishers, 1975

[A-3] William H. Press et. al. *Numerical Recipes*. New York: Cambridge University Press, 1986

1. The number in parentheses is a CNA internal control number.

APPENDIX B

SOURCE CODE

```
      PARAMETER (NMAX=14)
C
      IMPLICIT REAL*8(A-H,O-Z)

      REAL*8 KAY,MACH,MTOFT,MU,MU0,MUTAU,MV,MG
      REAL*8 MACHCD,MACHCL,MCHAB,MCHMIL
C
      COMMON /AERO/AIRSPD(NMAX),AOAMAX(NMAX),AAMXSD(NMAX),
     2            CLFTCD(NMAX),MACHCD(NMAX),
     3            CD,CDRAG(NMAX,NMAX),CDRGSD(NMAX,NMAX),
     4            AOACL(NMAX),MACHCL(NMAX),
     5            CL,CLIFT(NMAX,NMAX),CLFTSD(NMAX,NMAX),
     6            DERCD,DCDDCL(NMAX,NMAX),DCDDCLSD(NMAX,NMAX),
     7            DERCL,DCLDA(NMAX,NMAX),DCLDASD(NMAX,NMAX)

      COMMON /AIRCFT/ACM,AREA,BINGO,WTAC,WTFLMX,WTFUEL

      COMMON /AOA/ALPHA,ALPHA0,AOATAU

      COMMON /CONSTS/G,COUNT,ICOUNT,MTOFT,P0,PI,RHO0,ROOT,RTODEG

      COMMON /CONTRL/CHIDOT,CHIDOT0,CHITAU,
     2              GMDOT,GMDOT0,GMTAU,
     3              VDOT,VDOT0,VTAU,
     4              DELTA,DT,GEES,KAY

      COMMON /CYCLE/MV,MG,REFOR,SNG,CSG,SNCHI,CSCHI,CT

      COMMON /FNCTN/F1,F2,F3,F4,F5,F6,F7

      COMMON /POWER/ALT(NMAX),SPDMAX(NMAX),SPMXSD(NMAX),
     2             ALTMIL(NMAX),MCHMIL(NMAX),
     3             PWRMIL(NMAX,NMAX),PRMLSD(NMAX,NMAX),
     4             FULMIL(NMAX,NMAX),FLMLSD(NMAX,NMAX),
     5             ALTAB(NMAX),MCHAB(NMAX),
     6             PWRAB(NMAX,NMAX),PRABSD(NMAX,NMAX),
     7             FULAB(NMAX,NMAX),FLABSD(NMAX,NMAX),
     8             PCTRPM(NMAX),TSFC(NMAX),TSFCSD(NMAX),
     9             MACH,TH,THMAX

      COMMON /ROLL/MU,MU0,MUTAU

      COMMON /TBLMAX/NMCHMX,NAOAMX,NALTML,NMCHML,NALTAB,NMCHAB,
     2              NCLCD,NMCHCD,NAOACL,NMCHCL,NCTRPM
```

```
      PROGRAM MOFO
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                             C
C     PROGRAM:  MODEL OF FLYING OBJECTS (MOFO)                                C
C                                                                             C
C     AUTHORS:  Michael W. Price                                              C
C               James E. Schliessmann                                         C
C                                                                             C
C               CENTER FOR NAVAL ANALYSES                                     C
C               4401 FORD AVENUE                                              C
C               P.O. BOX 16268                                                C
C               ALEXANDRIA, VA 22302-0268                                     C
C               (703) 824-2693                                                C
C                                                                             C
C          PROGRAM MOFO IS THE MAIN ROUTINE FOR THE MOFO MODEL.  IT FIRST     C
C     INITIALIZES SEVERAL OF THE VARIABLES USED THROUGHOUT THE REST OF THE    C
C     PROGRAM.  IT THEN CONTINUES ON TO READ IN THE AIRCRAFT DATA FILE.       C
C                                                                             C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      INCLUDE 'CMNBLK/LIST'
C
      INTEGER IALT(NMAX)
C
      DATA CHIDOT0/0.D0/
      DATA G/32.17D0/
      DATA GMDOT0/0.D0/
      DATA MTOFT/3.280840D0/
      DATA P0/2116.2D0/
      DATA PI/3.14159265358979D0/
      DATA RHO0/2.377D-3/
      DATA VDOT0/0.D0/
C
      ROOT = 1.D0/3.5D0
      RTODEG = 180.D0/PI

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                             C
C          BEGIN READING THE AIRCRAFT DATA FILE.  START BY READING IN THE     C
C     ARRAY OF MAXIMUM MACH NUMBER AS A FUNCTION OF ALTITUDE AND THE ARRAY    C
C     OF MAXIMUM ANGLE OF ATTACK AS A FUNCTION OF MACH.  FOR EACH ARRAY,      C
C     SUBROUTINE SPLINE IS CALLED TO CREATE AN ARRAY OF SECOND DERIVATIVES    C
C     TO BE USED LATER WHEN INTERPOLATING FOR SPECIFIC VALUES.                C
C                                                                             C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      READ(1,1001) NMCHMX,NAOAMX
C
      READ(1,1003) (IALT(I),I=1,NMCHMX)
      READ(1,1002) (SPDMAX(I),I=1,NMCHMX)
      DO 69 I = 1,NMCHMX
         ALT(I) = DFLOAT(IALT(I))
   69 CONTINUE
      CALL SPLINE(ALT,SPDMAX,NMCHMX,1.D30,1.D30,SPMXSD)
C
      READ(1,1002) (AIRSPD(I),I=1,NAOAMX)
      READ(1,1002) (AOAMAX(I),I=1,NAOAMX)
      DO 9696 I=1,NAOAMX
         AOAMAX(I)=AOAMAX(I)/RTODEG
```

```
 9696 CONTINUE
      CALL SPLINE(AIRSPD,AOAMAX,NAOAMX,1.D30,1.D30,AAMXSD)
C
      WRITE(7,3333)
      WRITE(8,3334)

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                            C
C          READ IN THE 2-DIMENSIONAL ARRAY OF MILITARY POWER AS A FUNCTION   C
C     OF ALTITUDE AND MACH NUMBER BY CALLING SUBROUTINE TABIN.  CALL         C
C     SUBROUTINE SPLIE2 TO CREATE AN ARRAY OF SECOND DERIVATIVES NEEDED      C
C     IN THE BICUBIC SPLINE INTERPOLATION ROUTINE.                           C
C                                                                            C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      CALL TABIN(ALTMIL,MCHMIL,PWRMIL,NALTML,NMCHML,2)
      CALL SPLIE2(ALTMIL,MCHMIL,PWRMIL,NALTML,NMCHML,PRMLSD)

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                            C
C          READ IN THE 2-DIMENSIONAL ARRAY OF AFTERBURNER POWER AS A         C
C     FUNCTION OF ALTITUDE AND MACH NUMBER.  CREATE AN ARRAY OF ITS SECOND   C
C     DERIVATIVES.                                                           C
C                                                                            C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      CALL TABIN(ALTAB,MCHAB,PWRAB,NALTAB,NMCHAB,2)
      CALL SPLIE2(ALTAB,MCHAB,PWRAB,NALTAB,NMCHAB,PRABSD)

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                            C
C          READ IN THE 2-DIMENSIONAL ARRAY OF FUEL CONSUMPTION UNDER         C
C     MILITARY POWER AS A FUNCTION OF ALTITUDE AND MACH NUMBER.  CREATE      C
C     AN ARRAY OF ITS SECOND DERIVATIVES.                                    C
C                                                                            C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      CALL TABIN(ALTMIL,MCHMIL,FULMIL,NALTML,NMCHML,2)
      CALL SPLIE2(ALTMIL,MCHMIL,FULMIL,NALTML,NMCHML,FLMLSD)

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                            C
C          READ IN THE 2-DIMENSIONAL ARRAY OF FUEL CONSUMPTION UNDER         C
C     AFTERBURNER POWER AS A FUNCTION OF ALTITUDE AND MACH NUMBER.  CREATE   C
C     AN ARRAY OF ITS SECOND DERIVATIVES.                                    C
C                                                                            C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      CALL TABIN(ALTAB,MCHAB,FULAB,NALTAB,NMCHAB,2)
      CALL SPLIE2(ALTAB,MCHAB,FULAB,NALTAB,NMCHAB,FLABSD)

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                            C
C          READ IN THE 2-DIMENSIONAL ARRAY OF DRAG COEFFICIENT AS A          C
C     FUNCTION OF LIFT COEFFICIENT AND MACH NUMBER.  CREATE AN ARRAY OF      C
C     ITS SECOND DERIVATIVES.  CALL SUBROUTINE PARDER TO CREATE AN ARRAY     C
C     OF PARTIAL DERIVATIVES WITH RESPECT TO THE LIFT COEFFICIENT FOR        C
C     EACH MACH NUMBER.  CREATE AN ARRAY OF ITS SECOND DERIVATIVES.          C
C                                                                            C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
      CALL TABIN(CLFTCD,MACHCD,CDRAG,NCLCD,NMCHCD,1)
      CALL SPLIE2(CLFTCD,MACHCD,CDRAG,NCLCD,NMCHCD,CDRGSD)
      CALL PARDER(CDRAG,CLFTCD,NCLCD,NMCHCD,DCDDCL)
      CALL SPLIE2(CLFTCD,MACHCD,DCDDCL,NCLCD,NMCHCD,DCDDCLSD)

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                              C
C          READ IN THE 2-DIMENSIONAL ARRAY OF LIFT COEFFICIENT AS A           C
C     FUNCTION OF ANGLE OF ATTACK AND MACH NUMBER.  CREATE AN ARRAY OF ITS     C
C     SECOND DERIVATIVES.  CREATE AN ARRAY OF ITS PARTIAL DERIVATIVES WITH     C
C     RESPECT TO ANGLE OF ATTACK. THEN CREATE AN ARRAY OF ITS SECOND           C
C     DERIVATIVES.                                                             C
C                                                                              C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      CALL TABIN(AOACL,MACHCL,CLIFT,NAOACL,NMCHCL,1)
      DO 6969 I=1,NAOACL
         AOACL(I)=AOACL(I)/RTODEG
 6969 CONTINUE
      CALL SPLIE2(AOACL,MACHCL,CLIFT,NAOACL,NMCHCL,CLFTSD)
      CALL PARDER(CLIFT,AOACL,NAOACL,NMCHCL,DCLDA)
      CALL SPLIE2(AOACL,MACHCL,DCLDA,NAOACL,NMCHCL,DCLDASD)

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                              C
C          READ IN THE ONE DIMENSIONAL TABLE OF THRUST SPECIFIC FUEL           C
C     CONSUMPTION AS A FUNCTION OF THE PERCENT OF RPM's.                       C
C                                                                              C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      READ(1,998) NCTRPM
      READ(1,1002) (PCTRPM(I),I=1,NCTRPM)
      READ(1,1002) (TSFC(I),I=1,NCTRPM)
      CALL SPLINE(PCTRPM,TSFC,NCTRPM,1.D30,1.D30,TSFCSD)

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                              C
C          READ IN THE DIFFERENTIAL TIME STEP SIZE, THE DELTA TIME STEP        C
C     SIZE, AND THE NAVIGATION CONSTANT FOR PROPORTIONAL NAVIGATION.           C
C                                                                              C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      READ(1,999) DT,DELTA,KAY
      COUNT = 1.D0/DT
      ICOUNT = NINT(COUNT)

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                              C
C          READ DIFFERENTIAL TIME STEP SIZE AND AERODYNAMIC TIME CONSTANTS     C
C     FOR ANGLE OF ATTACK AND ROLL ANGLE FILTERING.                            C
C                                                                              C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      READ(1,1000) AOATAU,MUTAU,GMTAU,CHITAU,VTAU

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                              C
C          READ IN THE AIRCRAFT'S PLANFORM AREA, BASE WEIGHT, AND MAXIMUM      C
C     FUEL CAPACITY.  THEN CALL SUBROUTINE CONTROL                             C
```

```
C                                                                              C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      READ(1,2000) AREA,WTAC,WTFLMX
C
      CALL CONTROL
C
      STOP
C ***
  998 FORMAT(I10)
  999 FORMAT(/3F10.0)
 1000 FORMAT(/5F10.0)
 1001 FORMAT(/2I10)
 1002 FORMAT(10X,14F5.0)
 1003 FORMAT(10X,14I5)
 2000 FORMAT(/3F15.0)
 3333 FORMAT(1X, 't (s), X (m), Y (m), Z (m), V (m/s), BODY YAW (deg.)
     1, BODY PITCH (deg.), BODY ROLL (deg.)')
 3334 FORMAT(1X, 't (s), m (slugs), GAMMA (deg.), CHI (deg.),
     & AOA (deg.), MU (deg.), T (lbs.), Al (m/s**2),
     & Ap (m/s**2), Ay (m/s**2), At (m/s**2)')
C ***
      END
```

```
      SUBROUTINE SPLINE(X,Y,N,YP1,YPN,Y2)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                             C
C          GIVEN ARRAYS X AND Y OF LENGTH N CONTAINING A TABULATED FUNCTION,  C
C     AND GIVEN VALUES YP1 AND YPN FOR THE FIRST DERIVATIVE OF THE            C
C     INTERPOLATING FUNCTION AT POINTS 1 AND N, RESPECTIVELY, THIS ROUTINE    C
C     RETURNS AN ARRAY Y2 OF LENGTH N WHICH CONTAINS THE SECOND DERIVATIVES   C
C     OF THE INTERPOLATING FUNCTION AT THE TABULATED POINTS Xi.  IF YP1       C
C     AND/OR YPN ARE EQUAL TO 1.E30 OR LARGER, THE ROUTINE IS SIGNALLED TO    C
C     SET THE CORRESPONDING BOUNDARY CONDITION FOR A NATURAL SPLINE, WITH     C
C     ZERO SECOND DERIVATIVE ON THAT BOUNDARY.                                C
C                                                                             C
C     REFERENCE: NUMERICAL RECIPES, W.H. PRESS, et al.                        C
C                CAMBRIDGE U. PRESS,1986                                      C
C                                                                             C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      IMPLICIT REAL*8 (A-H,O-Z)
C
      PARAMETER (NMAX=14)
C
      REAL*8 X(NMAX),Y(NMAX),Y2(NMAX),U(NMAX)
C
      IF (YP1 .GT. .99D30) THEN
         Y2(1) = 0.D0
         U(1) = 0.D0
      ELSE
         Y2(1) = -0.5D0
         U(1) = (3.D0/(X(2)-X(1)))*((Y(2)-Y(1))/(X(2)-X(1))-YP1)
      ENDIF
C
      DO 11 I = 2,N-1
         SIG = (X(I)-X(I-1))/(X(I+1)-X(I-1))
         P = SIG*Y2(I-1)+2.D0
         Y2(I) = (SIG-1.D0)/P
         U(I) = (6.D0*((Y(I+1)-Y(I))/(X(I+1)-X(I))-(Y(I)-Y(I-1))
     *             /(X(I)-X(I-1)))/(X(I+1)-X(I-1))-SIG*U(I-1))/P
   11 CONTINUE
C
      IF (YPN .GT. .99D30) THEN
         QN = 0.D0
         UN = 0.D0
      ELSE
         QN = 0.5D0
         UN = (3.D0/(X(N)-X(N-1)))*(YPN-(Y(N)-Y(N-1))/(X(N)-X(N-1)))
      ENDIF
C
      Y2(N) = (UN-QN*U(N-1))/(QN*Y2(N-1)+1.D0)
C
      DO 12 K = N-1,1,-1
         Y2(K) = Y2(K)*Y2(K+1)+U(K)
   12 CONTINUE
C
      RETURN
      END


      SUBROUTINE SPLINT(XA,YA,Y2A,N,X,Y)
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                              C
C          GIVEN THE ARRAYS XA AND YA OF LENGTH N, WHICH TABULATE A FUNCTION  C
C     (WITH THE XAi's IN ORDER), AND GIVEN THE ARRAY Y2A, WHICH IS             C
C     THE OUTPUT FROM SPLINE ABOVE, AND GIVEN A VALUE OF X, THIS ROUTINE       C
C     RETURNS A CUBIC-SPLINE INTERPOLATED VALUE Y.                             C
C                                                                              C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      IMPLICIT REAL*8 (A-H,O-Z)
C
      PARAMETER (NMAX=14)
C
      REAL*8 XA(NMAX),YA(NMAX),Y2A(NMAX)
C
      KLO = 1
      KHI = N
C
      DO WHILE (KHI-KLO .GT. 1)
         K = (KHI+KLO)/2
         IF (XA(K) .GT. X) THEN
            KHI = K
         ELSE
            KLO = K
         ENDIF
      END DO
C
      H = XA(KHI) - XA(KLO)
      IF (H .EQ. 0.D0) THEN
         PAUSE 'Bad XA input.'
      ENDIF
C
      A = (XA(KHI)-X)/H
      B = (X-XA(KLO))/H
      Y = A*YA(KLO)+B*YA(KHI)+
     *          ((A**3-A)*Y2A(KLO)+(B**3-B)*Y2A(KHI))*(H**2)/6.D0
C
      RETURN
      END


      SUBROUTINE SPLIE2(X1A,X2A,YA,M,N,Y2A)

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                              C
C          GIVEN AN M BY N TABULATED FUNCTION YA, AND TABULATED INDEPENDENT    C
C     VARIABLES X1A (M VALUES) AND X2A (N VALUES), THIS ROUTINE CONSTRUCTS     C
C     ONE-DIMENSIONAL NATRUAL CUBIC SPLINES OF THE ROWS OF YA AND RETURNS      C
C     THE SECOND-DERIVATIVES IN THE ARRAY Y2A.                                 C
C                                                                              C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      IMPLICIT REAL*8 (A-H,O-Z)
C
      PARAMETER (NMAX=14)
C
      REAL*8 X1A(NMAX),X2A(NMAX),YA(NMAX,NMAX),Y2A(NMAX,NMAX),
     *          YTMP(NMAX),Y2TMP(NMAX)
C
      DO 13 J = 1,M
```

```
         DO 11 K = 1,N
            YTMP(K) = YA(J,K)
   11    CONTINUE
         CALL SPLINE(X2A,YTMP,N,1.D30,1.D30,Y2TMP)
         DO 12 K = 1,N
            Y2A(J,K) = Y2TMP(K)
   12    CONTINUE
   13 CONTINUE
C
      RETURN
      END


      SUBROUTINE SPLIN2(X1A,X2A,YA,Y2A,M,N,X1,X2,Y)

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                                C
C          GIVEN X1A, X2A, YA, M, N AS DESCRIBED IN SPLIE2 AND Y2A AS           C
C     PRODUCED BY THAT ROUTINE; AND GIVEN A DESIRED INTERPOLATING POINT X1,     C
C     X2; THIS ROUTINE RETURNS AN INTERPOLATED FUNCTION VALUE Y BY BICUBIC      C
C     SPLINE INTERPOLATION.                                                      C
C                                                                                C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      IMPLICIT REAL*8 (A-H,O-Z)
C
      PARAMETER (NMAX=14)
C
      REAL*8 X1A(NMAX),X2A(NMAX),YA(NMAX,NMAX),Y2A(NMAX,NMAX),
     *          YTMP(NMAX),Y2TMP(NMAX),YYTMP(NMAX)
C
      DO 12 J = 1,M
         DO 11 K = 1,N
            YTMP(K) = YA(J,K)
            Y2TMP(K) = Y2A(J,K)
   11    CONTINUE
         CALL SPLINT(X2A,YTMP,Y2TMP,N,X2,YYTMP(J))
   12 CONTINUE
C
      CALL SPLINE(X1A,YYTMP,M,1.D30,1.D30,Y2TMP)
      CALL SPLINT(X1A,YYTMP,Y2TMP,M,X1,Y)
C
      RETURN
      END
```

```
      SUBROUTINE TABIN(X1ARGS,X2ARGS,TABLE,NROWS,NCOLS,KFLAG)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                             C
C          SUBROUTINE TABIN IS CALLED FROM PROGRAM MOFO TO READ IN THE       C
C     2-DIMENSIONAL TABLES FROM THE AIRCRAFT DATA FILE.  DEPENDING ON THE     C
C     VALUE OF KFLAG, TABIN WILL READ IN EITHER INTEGER OR REAL VALUES.       C
C     IF VARIABLE SCALE IS NOT EQUAL TO ONE, THEN THE ARRAY IS MULTIPLIED     C
C     BY THIS FACTOR.                                                         C
C                                                                             C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      PARAMETER (NMAX=14)
C
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 TABLE(NMAX,NMAX),X1ARGS(NMAX),X2ARGS(NMAX)
C
      INTEGER ITABLE(NMAX,NMAX),IARGS(NMAX)
C
      READ(1,1000) NROWS,NCOLS,SCALE
      READ(1,1001) (X2ARGS(IC), IC=1,NCOLS)
      IF (KFLAG .EQ. 1) THEN
         DO 10 IR = 1,NROWS
            READ(1,1002)X1ARGS(IR),(TABLE(IR,IC),IC=1,NCOLS)
   10    CONTINUE
      ELSE IF (KFLAG .EQ. 2) THEN
         DO 20 IR = 1,NROWS
            READ(1,1003)IARGS(IR),(ITABLE(IR,IC),IC=1,NCOLS)
            X1ARGS(IR) = DFLOAT(IARGS(IR))
            DO 30 IC = 1,NCOLS
               TABLE(IR,IC) = DFLOAT(ITABLE(IR,IC))
   30       CONTINUE
   20    CONTINUE
      ENDIF
      IF (SCALE .NE. 1.D0) THEN
         DO 40 IR=1,NROWS
            DO 50 IC=1,NCOLS
               TABLE(IR,IC)=SCALE*TABLE(IR,IC)
   50       CONTINUE
   40    CONTINUE
      ENDIF
      RETURN
C ***
 1000 FORMAT(10X,2I5,30X,F10.0/)
 1001 FORMAT(10X,14F5.0)
 1002 FORMAT(4X,F5.0,1X,14F5.0)
 1003 FORMAT(4X,I5,1X,14I5)
C ***
      END
```

```
      SUBROUTINE PARDER(ARRAY,X,IMAX,JMAX,DERIV)

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                               C
C          SUBROUTINE PARDER IS CALLED FROM PROGRAM MOFO.  IT TAKES THE         C
C     VALUES IN ARRAY AND CALCULATES THE PARTIAL DERIVATIVES WITH RESPECT       C
C     TO VARIABLE ARRAY X.                                                      C
C                                                                               C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      PARAMETER (NMAX=14)
C
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 ARRAY(NMAX,NMAX),DERIV(NMAX,NMAX),X(NMAX)
C
      DO 500 J=1,JMAX
         DERIV(1,J) = (ARRAY(2,J)-ARRAY(1,J))/(X(2)-X(1))
         DERIV(IMAX,J) = (ARRAY(IMAX,J)-ARRAY(IMAX-1,J))/
     &                   (X(IMAX)-X(IMAX-1))
         DO 300 I=2,IMAX-1
            DERIV(I,J) = (ARRAY(I+1,J)-ARRAY(I-1,J))/(X(I+1)-X(I-1))
  300    CONTINUE
  500 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE CONTROL
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                                C
C          SUBROUTINE CONTROL IS CALLED FROM PROGRAM MOFO.  ITS TASK IS TO       C
C     PROCESS THE FILE THAT COMMANDS THE AIRCRAFT TO PERFORM TO THE USER'S      C
C     SPECIFICATIONS.  ORDERS ARE READ IN AND THEN TRANSFERRED TO THE           C
C     APPRORIATE SUBROUTINE TO COMPLETE THE COMMAND.  AT THE END OF THE         C
C     COMMAND FILE, THE SUBROUTINE KICKS OUT THUS ENDING A MOFO SESSION.        C
C                                                                                C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      INCLUDE 'CMNBLK/LIST'
C
      REAL*8 MXMACH
C
      CHARACTER*2 ORDERS
C
      LOGICAL FLAG

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                                C
C          READ IN THE INITIAL CONDITIONS.                                       C
C                                                                                C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      READ(2,*)
      READ(2,*)
      READ(2,5) X0,Y0,Z0,V0
      READ(2,5) CHI0,GM0,MU0,ALPHA0
      READ(2,5) T0,WTFUEL,BINGO,WTARM
      READ(2,*)

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                                C
C          A CONVERSION TO MIELE'S COORDINATE SYSTEM IS PERFORMED.  A CHECK      C
C     ON FUEL CAPICITY IS MADE.  THE WEIGHTS OF THE VEHICLE, ITS FUEL, AND      C
C     EXTERNAL STORES ARE ADDED AND CONVERTED TO MASS.                          C
C                                                                                C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      X = X0*MTOFT
      Y = -Y0*MTOFT
      Z = -Z0*MTOFT
C
      V = V0*MTOFT
      GM = GM0/RTODEG
      CHI = CHI0/RTODEG - PI/2.D0
C
      T = T0
      ALPHA0 = ALPHA0/RTODEG
      ALPHA = ALPHA0
      MU0 = MU0/RTODEG
      MU = MU0
C
      IF (WTFUEL .GT. WTFLMX) THEN
         WRITE(6,*) 'WARNING:'
         WRITE(6,*) ' THE FUEL WEIGHT IS GREATER THAN ALLOWED !!'
         WRITE(6,*) ' IT HAS BEEN RESET TO THE MAXIMUM FUEL WEIGHT.'
         WRITE(9,*) 'WARNING:'
```

```
            WRITE(9,*) ' THE FUEL WEIGHT IS GREATER THAN ALLOWED !!'
            WRITE(9,*) ' IT HAS BEEN RESET TO THE MAXIMUM FUEL WEIGHT.'
            WTFUEL = WTFLMX
         ENDIF
C
         IF (WTFUEL .LT. BINGO) THEN
            WRITE(6,*) 'WARNING:'
            WRITE(6,*) ' THE FUEL WEIGHT IS LESS THAN THE BINGO FUEL !!'
            WRITE(9,*) 'WARNING:'
            WRITE(9,*) ' THE FUEL WEIGHT IS LESS THAN THE BINGO FUEL !!'
         ENDIF
C
         IF (WTFUEL .LT. 0.D0) THEN
            WRITE(6,*) 'WARNING:'
            WRITE(6,*) ' FUEL WEIGHT IS LESS THAN ZERO !!'
            WRITE(9,*) 'WARNING:'
            WRITE(9,*) ' FUEL WEIGHT IS LESS THAN ZERO !!'
         ENDIF
C
         ACM = (WTAC + WTFUEL + WTARM)/G
C
         H = -Z
         ROH = RHO(H)
         MACH = V/DSQRT(1.4D0*P(H)/ROH)
         CALL SPLINT(ALT,SPDMAX,SPMXSD,NMCHMX,H,MXMACH)
         IF (MACH .GT. MXMACH) THEN
            WRITE(6,*) 'WARNING:'
            WRITE(6,*) ' INITIAL SPEED EXCEEDS MAXIMUM VELOCITY !!'
            WRITE(6,*) ' VALUE RESET TO MAXIMUM VELOCITY.'
            WRITE(9,*) 'WARNING:'
            WRITE(9,*) ' INITIAL SPEED EXCEEDS MAXIMUM VELOCITY !!'
            WRITE(9,*) ' VALUE RESET TO MAXIMUM VELOCITY.'
            MACH = MXMACH
            V = MACH * DSQRT(1.4D0*P(H)/ROH)
         ENDIF
C
         MG = ACM*G
         VSQ = V**2
         REFOR = 0.5D0*ROH*VSQ*AREA
         SNG = DSIN(GM)
         CALL SPLIN2(AOACL,MACHCL,CLIFT,CLFTSD,NAOACL,NMCHCL,
        &            ALPHA,MACH,CL)
         CALL SPLIN2(CLFTCD,MACHCD,CDRAG,CDRGSD,NCLCD,NMCHCD,CL,MACH,CD)
         TH = (MG*SNG + REFOR*CD)/DCOS(ALPHA)
C
         X1 = T
         X2 = X
         X3 = Y
         X4 = Z
         X5 = V
         X6 = GM
         X7 = CHI
         X8 = ALPHA
         X9 = MU
         CALL ATTITUDE(X1,X2,X3,X4,X5,X6,X7,X8,X9)

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                              C
C          COMMANDS ARE READ IN AND THE APPROPRIATE SUBROUTINES ARE CALLED.    C
C                                                                              C
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      FLAG = .TRUE.
      DO WHILE (FLAG)
         READ(2,10,ERR=999) ORDERS
         IF (ORDERS .EQ. 'WA') THEN
            CALL WAYPT(X,Y,Z,V,GM,CHI,T)
C        ELSE IF (ORDERS .EQ. 'HE') THEN
C           CALL HEADIN
C        ELSE IF (ORDERS .EQ. 'PI') THEN
C           CALL PITCH
C        ELSE IF (ORDERS .EQ. 'HP') THEN
C           CALL HDGPTC
C        ELSE IF (ORDERS .EQ. 'DR') THEN
C           CALL DRPSTR
C        ELSE IF (ORDERS .EQ. 'TF') THEN
C           CALL TERFOL
         ELSE IF (ORDERS .EQ. 'EN') THEN
            FLAG = .FALSE.
         ELSE
            WRITE(6,*) ' COMMAND ',ORDERS,' DOESN''T EXIST !!!'
            WRITE(9,*) ' COMMAND ',ORDERS,' DOESN''T EXIST !!!'
            FLAG = .FALSE.
         ENDIF
      END DO
C
  999 CONTINUE
      RETURN
C ***
    5 FORMAT(4F10.0)
   10 FORMAT(A)
C ***
      END
```

```
      SUBROUTINE ATTITUDE(X1,X2,X3,X4,X5,X6,X7,X8,X9)

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                              C
C          SUBROUTINE ATTITUDE IS CALLED ONCE FROM SUBROUTINE CONTROL. IT IS   C
C     CALLED THEREAFTER AT THE END OF THE RK SUBROUTINE.  ITS FUNCTION IS      C
C     TO COMPUTE THE BODY ATTITUDE ANGLES OF THE AIRCRAFT RELATIVE TO THE      C
C     INERTIAL REFERENCE FRAME.  IT THEN DUMPS OUT THE AIRCRAFT'S POSITION     C
C     SPEED AND BODY ATTITUDE EVERY SECOND.                                    C
C                                                                              C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      INCLUDE 'CMNBLK/LIST'
C
      REFOR = .5D0*RHO(-X4)*(X5**2)*AREA
C
      ALNG = (TH*DCOS(X8)-REFOR*CD)/ACM
      ALNG = ALNG-G*DSIN(X6)
C
      FL = REFOR*CL
      APTCH = ((FL+TH*DSIN(X8))*DCOS(X9)-ACM*G*DCOS(X6))/(ACM*X5)
C
      AYW = ((FL+TH*DSIN(X8))*DSIN(X9))/(ACM*X5*DCOS(X6))
C
      ALNG = ALNG/G
      APTCH = APTCH*X5/G
      AYW = AYW*X5*DCOS(X6)/G
      ATOTAL = DSQRT(ALNG**2+APTCH**2+AYW**2)

      X2 = X2/MTOFT
      X3 = -X3/MTOFT
      X4 = -X4/MTOFT
      X5 = X5/MTOFT
      X6 = X6
      X7 = X7
      X8 = X8
      CAL = DCOS(X8)
      SAL = DSIN(X8)
      CGM = DCOS(X6)
      SGM = DSIN(X6)
      CCH = DCOS(X7)
      SCH = DSIN(X7)
      CMU = DCOS(X9)
      SMU = DSIN(X9)
      AA = CAL*CGM*CCH-SAL*(CMU*SGM*CCH+SMU*SCH)
      BB = CAL*CGM*SCH-SAL*(CMU*SGM*SCH-SMU*CCH)
      CC = (-CAL*SGM-SAL*CMU*CGM)
      BCHI = DATAN(BB/AA)*RTODEG
      IF ((AA .LT. 0.D0) .AND. (BB .LE. 0.D0)) BCHI=180.D0+BCHI
      IF ((AA .LT. 0.D0) .AND. (BB .GT. 0.D0)) BCHI=180.D0+BCHI
      BCHI = 90.D0+BCHI
      BGM = DATAN(-CC/DSQRT(AA**2+BB**2))
      BMU = (CGM*SMU/DCOS(BGM))*RTODEG
      BGM = BGM*RTODEG
      X6 = X6*RTODEG
      X7 = X7*RTODEG
      X7 = X7+90.D0
      X8 = X8*RTODEG
      X9 = X9*RTODEG
      WRITE(7,3000)X1,X2,X3,X4,X5,BCHI,BGM,BMU
```

```
      WRITE(8,4000)X1,ACM,X6,X7,X8,X9,TH,ALNG,APTCH,AYW,ATOTAL
      RETURN
C ***
 3000 FORMAT(1X,5F11.2,3F11.4)
 4000 FORMAT(1X,11F10.2)
C ***
      END
```

```
      SUBROUTINE WAYPT(X,Y,Z,V,GM,CHI,T)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                              C
C          SUBROUTINE WAYPOINT IS CALLED FROM CONTROL TO CONSTRUCT A FLIGHT    C
C     SEGMENT.  IT IS PASSED THE PRESENT POSITION OF THE AIRCRAFT AND THE      C
C     DIRECTION OF THE VELOCITY VECTOR.  THE SUBROUTINE THEN READS IN ITS      C
C     COMMANDED POSITION AND CALCULATES THE DIRECTIONAL CHANGE NECESSARY TO    C
C     POINT THE VELOCITY VECTOR TO THE DESIRED WAYPOINT.  FINALLY, IT          C
C     CONSTRUCTS A FLIGHT PATH TO ACHIEVE THE DESIRED POSITION.                C
C                                                                              C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      INCLUDE 'CMNBLK/LIST'
C
      REAL*8 MXMACH
C
      LOGICAL FLAG

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                              C
C          READ IN THE DESTINATION WAYPOINT AND CONVERT TO MIELE'S             C
C     COORDINATE SYSTEM.                                                       C
C                                                                              C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      READ(2,10) XF,YF,ZF,VF,GEES
      READ(2,*)
C
      XF = XF*MTOFT
      YF = -YF*MTOFT
      ZF = -ZF*MTOFT
      VF = VF*MTOFT
C
      H = -ZF
      ROH = RHO(H)
      MACH = VF/DSQRT(1.4D0*P(H)/ROH)
      CALL SPLINT(ALT,SPDMAX,SPMXSD,NMCHMX,H,MXMACH)
      IF (MACH .GT. MXMACH) THEN
         WRITE(6,*) 'WARNING:'
         WRITE(6,*) ' COMMANDED SPEED EXCEEDS MAXIMUM VELOCITY !!'
         WRITE(6,*) ' VALUE RESET TO MAXIMUM VELOCITY.'
         WRITE(9,*) 'WARNING:'
         WRITE(9,*) ' COMMANDED SPEED EXCEEDS MAXIMUM VELOCITY !!'
         WRITE(9,*) ' VALUE RESET TO MAXIMUM VELOCITY.'
         MACH = MXMACH
         VF = MACH * DSQRT(1.4D0*P(H)/ROH)
      ENDIF

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                              C
C          BEGIN A DO LOOP TO COMMAND THE VELOCITY VECTOR INTO THE DIRECTION   C
C     OF THE WAYPOINT.  INTEGRATION IS PERFORMED UNTIL THE DIFFERENCE          C
C     BETWEEN THE COMMANDED VELOCITY VECTOR AND THE CURRENT VECTOR IS          C
C     MINIMAL.                                                                 C
C                                                                              C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      FLAG = .TRUE.
      DO WHILE (FLAG)
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                              C
C          COMPUTE THE RANGE AND THE GROUND RANGE BETWEEN THE AIRCRAFT'S       C
C     PRESENT POSITION AND ITS DESTINATION.                                    C
C                                                                              C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

         DELX = XF - X
         DELY = YF - Y
         DELZ = ZF - Z
C
         VX = V*DCOS(GM)*DCOS(CHI)
         VY = V*DCOS(GM)*DSIN(CHI)
         VZ = -V*DSIN(GM)
C
         RANGE = DSQRT(DELX**2 + DELY**2 + DELZ**2)
C
         OMGALOSX = -(DELY*VZ-DELZ*VY)/RANGE**2
         OMGALOSY = -(DELZ*VX-DELX*VZ)/RANGE**2
         OMGALOSZ = -(DELX*VY-DELY*VX)/RANGE**2
C
         DELV = VF - V
         AVEV = (VF + V)/2.D0
         TTI = RANGE/AVEV
         VDOT = DELV/TTI
         DRAGACCEL = (CD*REFOR/ACM)+G*DSIN(GM)
         IF ((VDOT .LT. 0.D0) .AND. (-VDOT .GT. DRAGACCEL)) THEN
            VDOT = DSIGN(DRAGACCEL,VDOT)
         ENDIF

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                              C
C          THE PITCH AND HEADING CHANGES THAT WILL BE NEEDED TO ACHIEVE THE    C
C     WAYPOINT ARE CALCULATED.                                                 C
C                                                                              C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

         IF (DSIN(CHI) .EQ. 0.D0) THEN
            GMDOT = KAY*(OMGALOSY-OMGALOSZ*DTAN(GM)*DSIN(CHI))/DCOS(CHI)
         ELSE
            GMDOT = KAY*(OMGALOSZ*DTAN(GM)*DCOS(CHI)-OMGALOSX)/DSIN(CHI)
         ENDIF
C
         CHIDOT =  KAY * OMGALOSZ/((DCOS(GM))**2)

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                              C
C          COMPUTE YAW AND PITCH ACCELERATIONS AT THESE COMMANDED RATES.       C
C                                                                              C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

         RATIO = DSQRT((GMDOT**2+(CHIDOT*DCOS(GM))**2)*V**2
     1             +VDOT**2)/(GEES*G)
         IF (RATIO .GT. 1.0D0) THEN
            CHIDOT = CHIDOT/RATIO
            GMDOT = GMDOT/RATIO
            VDOT = VDOT/RATIO
         ENDIF
C
```

```
         CHIDOTI = CHIDOT
         CHIDOT = FCHIDOT(CHIDOTI)
         GMDOTI = GMDOT
         GMDOT = FGMDOT(GMDOTI)
         VDOTI = VDOT
         VDOT = FVDOT(VDOTI)

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                              C
C          INTEGRATION IS PERFORMED IF THE DIFFERENCES WARRANT IT.             C
C                                                                              C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

         TF = T + DELTA
         DO WHILE ((NINT(T*COUNT) .LT. NINT(TF*COUNT)) .AND. FLAG)
            CALL AERO(X,Y,Z,V,GM,CHI,T)
            CALL RK(X,Y,Z,V,GM,CHI,T)
            DELX = XF - X
            DELY = YF - Y
            DELZ = ZF - Z
            RANGE = DSQRT(DELX**2 + DELY**2 + DELZ**2)
            IF (RANGE .LT. 328.D0) FLAG=.FALSE.
         END DO
      END DO
      RETURN
C ***
   10 FORMAT(5F10.0)
C ***
      END
```

```
      DOUBLE PRECISION FUNCTION FALPHA(ALPHAI)

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                              C
C          FUNCTION FALPHA IS CALLED FROM SUBROUTINE AERO.  THIS FUNCTION'S    C
C     JOB IS TO FILTER THE ANGLE OF ATTACK.  A 1ST-ORDER LINEAR FILTER IS      C
C     USED IN THE CALCULATION.  ALPHAI, THE PASSED PARAMETER, IS THE           C
C     COMMANDED ANGLE OF ATTACK.  ALPHA0 IS THE PRESENT HELD VALUE OF ANGLE    C
C     OF ATTACK.                                                               C
C                                                                              C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      INCLUDE 'CMNBLK/LIST'
C
      TERM = DEXP(-DT/AOATAU)
      FALPHA = ALPHA0*TERM + ALPHAI*(1.D0-TERM)
      ALPHA0 = FALPHA
      RETURN
      END
CCCCC
CCCCC
CCCCC
      DOUBLE PRECISION FUNCTION FMU(MUI)

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                              C
C          FUNCTION FMU IS CALLED FROM SUBROUTINE AERO.  THIS FUNCTIONS'S      C
C     JOB IS TO FILTER THE ROLL ANGLE.  A 1ST-ORDER LINEAR FILTER IS USED      C
C     IN THE CALCULATION.  MUI, THE PASSED PARAMETER, IS THE COMMANDED ROLL    C
C     ANGLE.  MU0 IS THE PRESENT HELD ROLL ANGLE.                              C
C                                                                              C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      INCLUDE 'CMNBLK/LIST'
C
      REAL*8 MUI
C
      TERM = DEXP(-DT/MUTAU)
      FMU = MU0*TERM + MUI*(1.D0-TERM)
      MU0 = FMU
      RETURN
      END
CCCCC
CCCCC
CCCCC
      DOUBLE PRECISION FUNCTION FGMDOT(GMDOTI)

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                              C
C          FUNCTION FGMDOT IS CALLED FROM VARIOUS SUBROUTINES.  THIS           C
C     FUNCTIONS'S JOB IS TO FILTER THE PITCH ACCELERATION.  A 1ST-ORDER        C
C     LINEAR FILTER IS USED IN THE CALCULATION.  GMDOTI, THE PASSED            C
C     PARAMETER, IS THE COMMANDED PITCH ANGLE.  GMDOT0 IS THE PRESENT HELD     C
C     PITCH ANGLE.                                                             C
C                                                                              C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      INCLUDE 'CMNBLK/LIST'
C
      TERM = DEXP(-DELTA/GMTAU)
```

```
      FGMDOT = GMDOTO*TERM + GMDOTI*(1.D0-TERM)
      GMDOTO = FGMDOT
      RETURN
      END
CCCCC
CCCCC
CCCCC
      DOUBLE PRECISION FUNCTION FCHIDOT(CHIDOTI)

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                             C
C          FUNCTION FCHIDOT IS CALLED FROM VARIOUS SUBROUTINES.  THIS         C
C     FUNCTIONS'S JOB IS TO FILTER THE YAW ACCELERATION.  A 1ST-ORDER         C
C     LINEAR FILTER IS USED IN THE CALCULATION.  CHIDOTI, THE PASSED          C
C     PARAMETER, IS THE COMMANDED YAW ANGLE.  CHIDOTO IS THE PRESENT          C
C     HELD YAW ANGLE.                                                         C
C                                                                             C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      INCLUDE 'CMNBLK/LIST'
C
      TERM = DEXP(-DELTA/CHITAU)
      FCHIDOT = CHIDOTO*TERM + CHIDOTI*(1.D0-TERM)
      CHIDOTO = FCHIDOT
      RETURN
      END
CCCCC
CCCCC
CCCCC
      DOUBLE PRECISION FUNCTION FVDOT(VDOTI)

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                             C
C          FUNCTION FVDOT IS CALLED FROM VARIOUS SUBROUTINES.  THIS           C
C     FUNCTIONS'S JOB IS TO FILTER THE LONGITUDNAL ACCELERATION.  A           C
C     1ST-ORDER LINEAR FILTER IS USED IN THE CALCULATION.  VDOTI,THE          C
C     PASSED PARAMETER, IS THE COMMANDED ACCELERATION.  VDOTO IS THE PRESENT  C
C     HELD ACCELERATION.                                                      C
C                                                                             C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      INCLUDE 'CMNBLK/LIST'
C
      TERM = DEXP(-DELTA/VTAU)
      FVDOT = VDOTO*TERM + VDOTI*(1.D0-TERM)
      VDOTO = FVDOT
      RETURN
      END
CCCCC
CCCCC
CCCCC

      DOUBLE PRECISION FUNCTION RHO(H)

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                             C
C          FUNCTION RHO COMPUTES THE STANDARD ATMOSPHERIC DENSITY AS A         C
C     FUNCTION OF ELEVATION, H.  THE PASSED PARAMETER H HAS UNITS FT.         C
C     THE DENSITY HAS UNITS SLUGS/FT**3.                                      C
C                                                                             C
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      INCLUDE 'CMNBLK/LIST'
C
      IF (H .GT. 3.6089D4) THEN
         RHO = RHO0*0.2971D0*DEXP((3.6089D4-H)/2.08067D4)
      ELSE
         THETA = 1.D0-(6.875D-6)*H
         RHO = RHO0*DEXP(4.2561D0*DLOG(THETA))
      ENDIF
      RETURN
      END
CCCCC
CCCCC
CCCCC
      DOUBLE PRECISION FUNCTION P(H)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                              C
C          FUNCTION P(H) COMPUTES THE STANDARD PRESSURE AS A FUNCTION OF       C
C     ELEVATION.  THE PASSED PARAMETER H HAS UNITS FT.  THE PRESSURE           C
C     HAS UNITS LBS/FT**2.                                                     C
C                                                                              C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      INCLUDE 'CMNBLK/LIST'
C
      IF (H .GT. 3.6089D4) THEN
         P = P0*0.2234D0*DEXP((3.6089D4-H)/2.08067D4)
      ELSE
         THETA = 1.D0-(6.875D-6)*H
         P = P0*DEXP(5.2561D0*DLOG(THETA))
      ENDIF
      RETURN
      END
```

```
      SUBROUTINE AERO(X,Y,Z,V,GM,CHI,T)

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                              C
C          SUBROUTINE AERO IS CALLED BY THE VARIOUS CONTROL ROUTINES.  ITS     C
C     FUNCTION IS TO CALCULATE THE AERODYNAMIC FORCES BASED ON THE             C
C     COMMANDED ACCELERATIONS.  THE CONTROL VARIABLES CALCULATED ARE THE       C
C     ANGLE OF ATTACK AND THE VELOCITY ROLL ANGLE.  THE THRUST NEEDED IS ALSO  C
C     CALCULATED.                                                              C
C                                                                              C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      INCLUDE 'CMNBLK/LIST'
C
      REAL*8 MUI
C
      H = -Z
      ROH = RHO(H)
      MACH = V/DSQRT(1.4D0*P(H)/ROH)
      MG = ACM*G
      MV = ACM*V
      VSQ = V**2
      REFOR = 0.5D0*ROH*VSQ*AREA
      SNG = DSIN(GM)
      CSG = DCOS(GM)
      SNCHI = DSIN(CHI)
      CSCHI = DCOS(CHI)

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                              C
C          COMPUTE THE ROLL ANGLE FROM THE COMMANDED PITCH AND YAW RATES.      C
C                                                                              C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      MUI = ATAN2((CHIDOT*CSG),(GMDOT+G*CSG/V))

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                              C
C          FILTER THE ROLL ANGLE.                                              C
C                                                                              C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      MU = FMU(MUI)

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                              C
C          CALL NEWTON TO DETERMINE THE ANGLE OF ATTACK CORRESPONDING TO THE  C
C     FILTERED ROLL ANGLE, COMMANDED PITCH RATE AND LONGITUDINAL ACCELERATION.C
C                                                                              C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      ALPHAI = ALPHA
      CALL NWTN1D(ALPHAI)

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                              C
C          INTERPOLATE TO FIND THE MAXIMUM ANGLE OF ATTACK AT THIS MACH        C
C     NUMBER.                                                                  C
C                                                                              C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
      IF (DABS(ALPHAI) .GT. 2.D0*PI) THEN
         ALPHAI = DMOD(ALPHAI,2.D0*PI)
      ENDIF
      IF (DABS(ALPHAI) .GT. PI) THEN
         TEMALF = 2.D0*PI - DABS(ALPHAI)
         ALPHAI = DSIGN(TEMALF,-ALPHAI)
      ENDIF
C
      CALL SPLINT(AIRSPD,AOAMAX,AAMXSD,NAOAMX,MACH,ALFMAX)
      IF (ALPHAI .GT. ALFMAX) THEN
         ALPHAI = ALFMAX
      ENDIF

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                              C
C          FILTER THE ANGLE OF ATTACK.                                         C
C                                                                              C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      ALPHA = FALPHA(ALPHAI)

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                              C
C          DETERMINE THE LIFT COEFFICIENT AT THIS VALUE OF ANGLE OF ATTACK.    C
C                                                                              C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      CALL SPLIN2(AOACL,MACHCL,CLIFT,CLFTSD,NAOACL,NMCHCL,
     &            ALPHA,MACH,CL)

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                              C
C          DETERMINE THE DRAG COEFFICIENT AT THIS LIFT COEFFICIENT.            C
C                                                                              C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      CALL SPLIN2(CLFTCD,MACHCD,CDRAG,CDRGSD,NCLCD,NMCHCD,CL,MACH,CD)

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                              C
C          DETERMINE THE MAXIMUM THRUST AT THIS ALTITUDE AND MACH NUMBER.      C
C                                                                              C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      CALL SPLIN2(ALTMIL,MCHMIL,PWRMIL,PRMLSD,NALTML,NMCHML,
     &            H,MACH,THMAX)

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                              C
C          COMPUTE THE THRUST FOR THIS COMMANDED LONGITUDINAL ACCELERATION.    C
C                                                                              C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      TH = (ACM*VDOT+MG*SNG+CD*REFOR)/DCOS(ALPHA)

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                              C
C          LIMIT THE THRUST.                                                   C
C                                                                              C
```

```
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

      IF (TH .GT. THMAX) THEN
         TH=THMAX
      ENDIF

cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                              c
c          COMPUTE THE PERCENT MAXIMUM RPMs, AND DETERMINE THE THRUST          c
c     SPECIFIC FUEL CONSUMPTION.                                               c
c                                                                              c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

      RPM = (TH/THMAX)**ROOT
      CALL SPLINT(PCTRPM,TSFC,TSFCSD,NCTRPM,RPM,CT)
c
      RETURN
      END
```

```
      SUBROUTINE NWTN1D(X)

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                              C
C          SUBROUTINE NEWTON PERFORMS A 1-DIMENSIONAL ROOT FINDING ALGORITHM  C
C     USING THE NEWTON-RAPHSON METHOD.                                         C
C                                                                              C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      IMPLICIT REAL*8 (A-H,O-Z)
C
      LOGICAL FLAG
C
      EPSLNX = .00875D0
      FLAG = .TRUE.
      JCOUNT = 0
      DO WHILE (FLAG)
C
         FUNC1 = FUNCF1(X)
C
         DF1X = FDF1X(X)
C
         DET = DF1X
C
         DETINV = 1.D0/DET
C
         XNEW = X - DETINV*FUNC1
C
         IF (DABS(X-XNEW) .LE. EPSLNX) THEN
            FLAG = .FALSE.
         ENDIF
C
         JCOUNT = JCOUNT + 1
         IF (JCOUNT .EQ. 50) THEN
            XNEW = (XNEW + X)/2.D0
         ELSE IF (JCOUNT .EQ. 1000) THEN
            WRITE(6,*) ' NEWTON OVERLOAD !!'
            WRITE(9,*) ' NEWTON OVERLOAD !!'
            STOP ' MAJOR ERROR !!'
         ENDIF
C
         X = XNEW
C
      END DO
C
      RETURN
      END


      DOUBLE PRECISION FUNCTION FUNCF1(ALPHAI)
C
      INCLUDE 'CMNBLK/LIST'
C
      CALL SPLIN2(AOACL,MACHCL,CLIFT,CLFTSD,NAOACL,NMCHCL,
     &              ALPHAI,MACH,CL)
      CALL SPLIN2(CLFTCD,MACHCD,CDRAG,CDRGSD,NCLCD,NMCHCD,CL,MACH,CD)
C
      FUNCF1 = ((((ACM*VDOT+MG*SNG+REFOR*CD)*DTAN(ALPHAI)+REFOR*CL)
     &             *DCOS(MU) - MG*CSG) / MV) - GMDOT
      RETURN
```

```
      END

      DOUBLE PRECISION FUNCTION FDF1X(ALPHAI)
C
      INCLUDE 'CMNBLK/LIST'
C
      CALL SPLIN2(AOACL,MACHCL,DCLDA,DCLDASD,NAOACL,NMCHCL,
     &            ALPHAI,MACH,DERCL)
      CALL SPLIN2(CLFTCD,MACHCD,DCDDCL,DCDDCLSD,NCLCD,NMCHCD,
     &            CL,MACH,DERCD)
C
      FDF1X = ( REFOR*CD*(1.D0/DCOS(ALPHAI))**2 +
     &          REFOR*DERCD*DERCL*DTAN(ALPHAI)  +
     &          REFOR*DERCL ) * DCOS(MU) / MV
      FDF1X = ((ACM*VDOT + MG*SNG)*(1.D0/DCOS(ALPHAI))**2)
     &        *(DCOS(MU)/MV) + FDF1X
      RETURN
      END
```

```
      SUBROUTINE RK(X,Y,Z,V,GM,CHI,T)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                              C
C          SUBROUTINE RK IS CALLED FROM THE VARIOUS CONTROL ROUTINES.  ITS     C
C     FUNCTION IS TO IMPLEMENT THE 4TH-ORDER RUNGE-KUTTA FORMULA.  SUBROUTINE C
C     RK ADVANCES A TIME-STEPPED SOLUTION TO THE DIFFERENTIAL SYSTEM          C
C     DESCRIBING THE TIME RATE OF CHANGE OF THE AIRCRAFT'S INERTIAL POSITION  C
C     AND VELOCITY, AND MASS.                                                  C
C                                                                              C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      INCLUDE 'CMNBLK/LIST'
C
      CALL F(X,Y,Z,V,GM,CHI,ACM,T)
C
      F11 = F1
      F21 = F2
      F31 = F3
      F41 = F4
      F51 = F5
      F61 = F6
      F71 = F7
      XSIN = X+DT*0.5D0*F11
      YSIN = Y+DT*0.5D0*F21
      ZSIN = Z+DT*0.5D0*F31
      VSIN = V+DT*0.5D0*F41
      GMSIN = GM+DT*0.5D0*F51
      CHSIN = CHI+DT*0.5D0*F61
      ACMSIN = ACM+DT*0.5D0*F71
      T = T+DT*0.5D0
      CALL F(XSIN,YSIN,ZSIN,VSIN,GMSIN,CHSIN,ACMSIN,T)
C
      F12 = F1
      F22 = F2
      F32 = F3
      F42 = F4
      F52 = F5
      F62 = F6
      F72 = F7
      XSSIN = X+DT*0.5D0*F12
      YSSIN = Y+DT*0.5D0*F22
      ZSSIN = Z+DT*0.5D0*F32
      VSSIN = V+DT*0.5D0*F42
      GMSSIN = GM+DT*0.5D0*F52
      CHSSIN = CHI+DT*0.5D0*F62
      ACMSSIN = ACM+DT*0.5D0*F72
      CALL F(XSSIN,YSSIN,ZSSIN,VSSIN,GMSSIN,CHSSIN,ACMSSIN,T)
C
      F13 = F1
      F23 = F2
      F33 = F3
      F43 = F4
      F53 = F5
      F63 = F6
      F73 = F7
      XSP1 = X+DT*F13
      YSP1 = Y+DT*F23
      ZSP1 = Z+DT*F33
      VSP1 = V+DT*F43
```

```
      GMSP1 = GM+DT*F53
      CHSP1 = CHI+DT*F63
      ACMSP1 = ACM+DT*F73
      T = T+DT*0.5D0
      CALL F(XSP1,YSP1,ZSP1,VSP1,GMSP1,CHSP1,ACMSP1,T)
C
      F14 = F1
      F24 = F2
      F34 = F3
      F44 = F4
      F54 = F5
      F64 = F6
      F74 = F7
      X = X+DT*(F11/6.D0+F12/3.D0+F13/3.D0+F14/6.D0)
      Y = Y+DT*(F21/6.D0+F22/3.D0+F23/3.D0+F24/6.D0)
      Z = Z+DT*(F31/6.D0+F32/3.D0+F33/3.D0+F34/6.D0)
      V = V+DT*(F41/6.D0+F42/3.D0+F43/3.D0+F44/6.D0)
      GM = GM+DT*(F51/6.D0+F52/3.D0+F53/3.D0+F54/6.D0)
      CHI = CHI+DT*(F61/6.D0+F62/3.D0+F63/3.D0+F64/6.D0)
      ACM = ACM+DT*(F71/6.D0+F72/3.D0+F73/3.D0+F74/6.D0)
      IF(CHI.LT.-PI) CHI=CHI+2.D0*PI
      IF(CHI.GT.PI) CHI=CHI-2.D0*PI
C
      JFLAG = NINT(T*COUNT)
      IF (MOD(JFLAG,ICOUNT) .EQ. 0) THEN
         X1 = T
         X2 = X
         X3 = Y
         X4 = Z
         X5 = V
         X6 = GM
         X7 = CHI
         X8 = ALPHA
         X9 = MU
         CALL ATTITUDE(X1,X2,X3,X4,X5,X6,X7,X8,X9)
      ENDIF
C
      RETURN
      END
```

```
      SUBROUTINE F(X,Y,Z,V,GM,CHI,MAC,T)

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                              C
C          SUBROUTINE F COMPUTES THE INSTANTANEOUS TIME RATES OF CHANGE OF     C
C     THE VEHICLE'S CARTESIAN POSITION, SPEED, VELOCITY PITCH AND YAW ANGLES   C
C     AND MASS. SUBROUTINE F IS CALLED BY SUBROUTINE RK.                       C
C                                                                              C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      INCLUDE 'CMNBLK/LIST'
C
      REAL*8 MAC
C
      H = -Z
      ROH = RHO(H)
      MG = MAC*G
      MV = MAC*V
      VSQ = V**2
      REFOR = 0.5D0*ROH*VSQ*AREA
      SNG = DSIN(GM)
      CSG = DCOS(GM)
      SNCHI = DSIN(CHI)
      CSCHI = DCOS(CHI)

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                              C
C          COMPUTE THE COMPONENT OF VELOCITY ALONG THE X-AXIS IN MIELE'S       C
C     FLAT EARTH COORDINATE SYSTEM.                                            C
C                                                                              C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      F1 = V*CSG*CSCHI

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                              C
C          COMPUTE THE COMPONENT OF VELOCITY ALONG THE Y-AXIS IN MIELE'S       C
C     FLAT EARTH COORDINATE SYSTEM.                                            C
C                                                                              C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      F2 = V*CSG*SNCHI

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                              C
C          COMPUTE THE COMPONENT OF VELOCITY ALONG THE Z-AXIS IN MIELE'S       C
C     FLAT EARTH COORDINATE SYSTEM.                                            C
C                                                                              C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      F3 = -V*SNG

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                              C
C          COMPUTE THE LONGITUDINAL ACCELERATION  V-DOT.                       C
C                                                                              C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      F4 = (TH*DCOS(ALPHA)-REFOR*CD)/MAC
      F4 = F4-G*SNG
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                              C
C          COMPUTE THE RATE OF CHANGE OF THE VELOCITY PITCH ANGLE GAMMA-DOT.  C
C                                                                              C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      FL = REFOR*CL
      F5 = ((FL+TH*DSIN(ALPHA))*DCOS(MU)-MAC*G*CSG)/(MAC*V)

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                              C
C          COMPUTE THE RATE OF CHANGE OF THE VELOCITY YAW ANGLE CHI-DOT.      C
C                                                                              C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      F6 = ((FL+TH*DSIN(ALPHA))*DSIN(MU))/(MAC*V*CSG)

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                              C
C          COMPUTE THE RATE OF CHANGE OF THE MASS.                             C
C                                                                              C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

       F7 = -CT*TH/(G*3600.D0)
C
      ALNG = F4/G
      APTCH = F5*V/G
      AYW = F6*V*CSG/G
      ATOTAL = DSQRT(ALNG**2+APTCH**2+AYW**2)
      IF (ATOTAL .GT. GEES) THEN
         WRITE(6,1111) T,ALNG,APTCH,AYW,ATOTAL
         WRITE(9,1111) T,ALNG,APTCH,AYW,ATOTAL
      ENDIF
      RETURN
C ***
 1004 FORMAT(1H ,'MIL')
 1005 FORMAT(1H ,'AB')
 1013 FORMAT(1X,'BINGO FUEL')
 1111 FORMAT(1H ,'FTIME,AL,AP,AY,AT=',5F10.4)
C ***
      END
```

APPENDIX C

MODMOFO.COM

APPENDIX C

MODMOFO.COM

This appendix contains the source code to the file MODMOFO.COM. It also contains a ".COM" file that establishes the common block cross-reference files. As alluded to earlier, the use of these files will make modifications to the source code much easier.

MODMOFO.COM is a derivative of the source code used in modifying another model within the Advanced Systems Division. The original code, MODALARM.COM, was authored by Ciro L. Pinto-Coelho. The user will find the source code internally documented for immediate implementation. The insightful user will find it applicable to other models as well.

```
$ !
$ !      This command file mimics the UNIX make utility in modifying
$ ! a particular group of source and common block files. This command
$ ! procedure relies on a specific directory structure.  The top level
$ ! directory has the same name as the model.  Below it are a number of
$ ! other subdirectories as follows:
$ !
$ !              [MODEL]
$ !                    |   --- Location of executable.
$ !              [MODEL.CODE]
$ !                      | --- Location of source code and this command
$ !                      |     procedure.
$ !              [MODEL.CODE.CMNBLKS]
$ !                           | --- Location of common block files.
$ !              [MODEL.CODE.CMNBLKS.DAT]
$ !                                   | --- Location of common block cross
$ !                                   |     reference files.
$ !
$ !      The procedure relies on the coding practice of include statements
$ ! to incorporate common blocks into a program.  The common blocks
$ ! are defined by files   whose prefix is the name of the common block -
$ ! in the subdirectory [MODEL.CODE.CMNBLKS].  These files have extension
$ ! ".CMN".  Further, each common block has a cross reference file -
$ ! located in the subdirectory [MODEL.CODE.CMNBLKS.DAT] - that contains
$ ! a list of those source code files that use it.  These files can be
$ ! generated with the command procedure MKDAT.COM in the subdirectory
$ ! [MODEL.CODE.CMNBLKS.DAT].
$ !
$ !      This proceudre relies on the fact that modified source or common
$ ! block files have version numbers greater than one.  Therefore,
$ ! this command procedure generates a list of the source code files that
$ ! have been modified and places them in a temporary file.  It then
$ ! generates a list of those common block files that have been modified.
$ ! These names are saved and the cross reference file for each common block
$ ! is read to determine which source ocde files must be compiled. From these
$ ! two steps the procedure builds an alphabetically sorted list of the
$ ! source code files to be recompiled.  The list is sorted to avoid
$ ! recompiling a source file more than once.  Each source file is then
$ ! recompiled and a copy is placed in the text library MODEL.TLB, and a copy
$ ! of the object file is placed in the library MODEL.OLB.  The linker is
$ ! then called to create an executable image name MODEL.EXE in the
$ ! subdirectory [MODEL].  After each successful compilation the procedure
$ ! purges the old version(s) of the source code file and revises its version
$ ! number to one. Recall that this is the key to the operation of this
$ ! procedure. Finally, a bit of clean-up of the temporary file is done before
$ ! completion.
$ !
$ !
$ ! Comments...
$ !
$ !     - An option was added to allow generation of listing files from the
$ !       compilation step.  The option is activated by supplying an "l" or
$ !       "L", for list, on the command-line.
$ !
$ ! Establish error handling procedures...
```

```
$ !
$ ON CONTROL_Y THEN GOTO FINISHED
$ ON ERROR THEN GOTO FINISHED
$ !
$ ! Define the model name and default file extension...
$ !
$ CMNDATPTH = "DISK4:[MOFO.CODE.CMNBLKS.DAT]"
$ CODEDIR   = "DISK4:[MOFO.CODE]"
$ CMNPTH    = "DISK4:[MOFO.CODE.CMNBLKS]"
$ MODEL     = "MOFO"
$ EXT       = "MFO"
$ !
$ IF P1 .NES. " " THEN P1 = f$edit(P1,"LOWERCASE")
$ !
$ ! Initialize the counter for files compiled...
$ !
$ NFILES = 0
$ !
$ ! Set the default to the code directory, clear the screen and begin
$ ! execution of the procedure...
$ !
$ SET DEFAULT 'CODEDIR'
$ SET TERMINAL/WIDTH=80
$ WRITE SYS$OUTPUT " "
$ WRITE SYS$OUTPUT MODEL," make utility..."
$ WRITE SYS$OUTPUT " "
$ !
$ ! Assign all common block files to logical names. This is a model specific
$ ! section of the command procedure...
$ !
$ ASSIGN   'CMNPTH'CMNBLK.CMN   CMNBLK
$ !
$ ! Generate a directory of those model routines that were modified.
$ ! Include those files that are affected by modifications to
$ ! their common block(s)...
$ !
$ WRITE SYS$OUTPUT "     Creating directory of modified source code files..."
$ DIRECTORY/VERSION=1/NOHEADING/NOTRAILING/OUTPUT='MODEL'.CODE/EXCLUDE=(*.*.1) *.'EXT'
$ !
$ WRITE SYS$OUTPUT "     Creating directory of modified common block files..."
$ DIRECTORY/VERSION=1/NOHEADING/NOTRAILING/OUTPUT='MODEL'.CMNS/EXCLUDE=(*.*.1) [.CMNBLKS]*.CMN
$ !
$ ! Generate a file that contains a list of the files affected
$ ! by the common modification...
$ !
$ WRITE SYS$OUTPUT "     Creating composite directory listing..."
$ !
$ OPEN/READ CMNS_FILE 'MODEL'.CMNS
$ OPEN/WRITE PURGE_CMNS_FILE 'MODEL'.PURGE
$ !
$ STARTCMNS:
$ !
$   READ/END_OF_FILE=ENDCMNS CMNS_FILE MODULE_NAME
$   MODULE_NAME = F$EDIT(MODULE_NAME,"COMPRESS,TRIM")
$   FILE = F$PARSE(MODULE_NAME,,,"NAME")
$   DFILE = CMNDATPTH + FILE + ".DAT"
```

```
$ !
$ ! Form one file that contains all the file names...
$ !
$   COPY 'MODEL'.CODE + 'DFILE' 'MODEL'.CODE
$ !
$   CFILE = CMNPTH + FILE + ".CMN"
$   LFILE = CMNPTH + MODEL + "CMN"
$ !
$   WRITE/SYMBOL PURGE_CMNS_FILE CFILE
$   LIBRARY/REPLACE/TEXT 'LFILE' 'CFILE'
$ !
$   GOTO STARTCMNS
$ !
$ ENDCMNS:
$ !
$ CLOSE PURGE_CMNS_FILE
$ !
$ ! Sort the modified file names so that the LAST_FILE construct
$ ! in the compilation loop will function properly...
$ !
$ WRITE SYS$OUTPUT "     Sorting composite directory..."
$ SORT 'MODEL'.CODE 'MODEL'.CODE
$ PURGE 'MODEL'.CODE
$ RENAME 'MODEL'.CODE.* 'MODEL'.CODE.1
$ !
$ ! Open the directory file for input...
$ !
$ OPEN/READ INPUT_FILE 'MODEL'.CODE
$ !
$ LAST_FILE = " "
$ !
$ ! Loop through the list of modified routines and compile each, being
$ ! careful not to compile a particular routine twice...
$ !
$ BEGIN:
$ !
$   READ/END_OF_FILE=END INPUT_FILE MODULE_NAME
$   MODULE_NAME = F$EDIT(MODULE_NAME,"COMPRESS,TRIM")
$   FILE = F$PARSE(MODULE_NAME,,,"NAME")
$ !
$   IF FILE .EQS. LAST_FILE THEN GOTO BEGIN
$ !
$   SOURCE = FILE + "." + EXT
$   OBJECT = FILE + ".OBJ"
$ !
$   SET TERMINAL/WIDTH=80
$   WRITE SYS$OUTPUT " "
$   WRITE SYS$OUTPUT "Recompiling module - ",f$edit(source,"LOWERCASE")
$ !
$ ! Determine if a list file is to be created...
$ !
$   IF P1 .EQS. "1" THEN GOTO LIST
$ !
$   FORTRAN/CONTINUATION=50/G_FLOATING/CHECK=ALL 'FILE'.'EXT'
$   GOTO CLEAN
$ !
```

```
$ LIST:
$   FORTRAN/CONTINUATION=50/G_FLOATING/LIST/CHECK=ALL 'FILE'.'EXT'
$ !
$ CLEAN:
$ !
$   PURGE 'FILE'.'EXT'
$   RENAME 'FILE'.'EXT'.*  'FILE'.'EXT'.1
$   NFILES = NFILES + 1
$ !
$   WRITE SYS$OUTPUT "Replacing text module - ",f$edit(source,"LOWERCASE")
$   LIB/REPLACE/TEXT 'MODEL' 'FILE'.'EXT'
$ !
$   WRITE SYS$OUTPUT "Replacing object module - ",f$edit(object,"LOWERCASE")
$   LIB/REPLACE/OBJ 'MODEL' 'FILE'.OBJ
$ !
$   DELETE 'FILE'.OBJ.*
$   LAST_FILE = FILE
$ !
$   GOTO BEGIN
$ !
$ END:
$ !
$ ! Compress the object library to minimize the amount of disk
$ ! space used...
$ !
$ LIB/COMPRESS/OBJ 'MODEL'
$ PURGE 'MODEL'.OLB
$ RENAME 'MODEL'.OLB 'MODEL'.OLB.1
$ !
$ ! Link the model to create an executable image if compilation has preceeded
$ ! this point...
$ !
$ IF NFILES .EQ. 0 THEN GOTO FINISHED
$ !
$    SET TERMINAL/WIDTH=80
$    WRITE SYS$OUTPUT " "
$    WRITE SYS$OUTPUT "Relinking the ",MODEL," model..."
$    LINK 'MODEL'/LIB/INCLUDE=('MODEL')/EXECUTABLE='MODEL'.EXE
$ !
$ ! Place the executable image in its default location...
$ !
$    DELETE [-]'MODEL'.EXE.*
$    RENAME 'MODEL'.EXE [-]'MODEL'.EXE.1
$ !
$    WRITE SYS$OUTPUT ""
$    WRITE SYS$OUTPUT MODEL," model modification completed!"
$    WRITE SYS$OUTPUT ""
$ !
$ !
$ ! Clean-up before exiting...
$ !
$    WRITE SYS$OUTPUT "Purging common block files..."
$    WRITE SYS$OUTPUT " "
$ !
$    OPEN/READ PURGE_CMNS_FILE 'MODEL'.PURGE
$ !
```

```
$     BEGINP:
$ !
$       READ/END_OF_FILE=ENDP PURGE_CMNS_FILE MODULE_NAME
$ !
$       PURGE 'MODULE_NAME'
$       RENAME 'MODULE_NAME'.* 'MODULE_NAME'.1
$       GOTO BEGINP
$ !
$     ENDP:
$ !
$ !
$ FINISHED:
$ !
$ ! Close and delete the directory file...
$ !
$ CLOSE INPUT_FILE
$ CLOSE CMNS_FILE
$ CLOSE PURGE_CMNS_FILE
$ !
$ DELETE 'MODEL'.CODE.*
$ DELETE 'MODEL'.CMNS.*
$ DELETE 'MODEL'.PURGE.*
$ !
$ ! Deassign all model specific logical symbols...
$ !
$ DEASSIGN   CMNBLK
$ !
$ WRITE SYS$OUTPUT "Done!"
$ EXIT
```

```
$ !
$ ! Command procedure to establish the common block cross reference files...
$ !
$ ON CONTROL_Y THEN GOTO FINISHED
$ ON ERROR THEN GOTO FINSHED
$ !
$ ! Define some logicals...
$ !
$ CMNDAT = "DISK4:[MOFO.CODE.CMNBLKS.DAT]"
$ CMNPTH = "DISK4:[MOFO.CODE.CMNBLKS]"
$ CODE   = "DISK4:[MOFO.CODE]"
$ MODEL  = "MOFO"
$ CEXT   = "CMN"
$ DEXT   = "DAT"
$ SEXT   = "MFO"
$ !
$ !
$ ! Generate a directory of those model common block files that were modified...
$ !
$ COMMONS = CMNPTH + "*." + CEXT
$ SFILE = CODE + "*." + SEXT
$ !
$ DIRECTORY/VERSION=1/NOHEADING/NOTRAILING/OUTPUT='MODEL'.DCT/EXCLUDE=(*.*.1) 'COMMONS'
$ !
$ ! Open the directory file for input...
$ !
$ OPEN/READ INPUT_FILE 'MODEL'.DCT
$ !
$ LAST_FILE = " "
$ !
$ ! Loop through the list of modified common blocks...
$ !
$ BEGIN:
$ !
$   READ/END_OF_FILE=END INPUT_FILE MODULE_NAME
$   MODULE_NAME = F$EDIT(MODULE_NAME,"COMPRESS,TRIM")
$   FILE = F$PARSE(MODULE_NAME,,,"NAME")
$ !
$   IF FILE .EQS. LAST_FILE THEN GOTO BEGIN
$ !
$   WRITE SYS$OUTPUT "  ",F$EDIT(FILE,"LOWERCASE"),"..."
$ !
$   OFILE = CMNDAT + FILE + "." + DEXT
$   SEARCH/WINDOW=0/OUTPUT='OFILE' 'SFILE' 'FILE'"/"LIST
$ !
$   LAST_FILE = FILE
$   GOTO BEGIN
$ !
$ END:
$ !
$ !
$ FINISHED:
$ !
$ ! Close and delete the directory file...
$ !
$ CLOSE INPUT_FILE
$ DELETE 'MODEL'.DCT.*
$ EXIT
```